1971

# Space dependent model for the slowing down of fast neutrons.

Mills, Thomas Martin.

Monterey, California ; Naval Postgraduate School

http://hdl.handle.net/10945/15908

# SPACE DEPENDENT MODEL
# FOR THE SLOWING DOWN OF FAST NEUTRONS

Thomas Martin Mills

United States
Naval Postgraduate School

# THESIS

SPACE DEPENDENT MODEL FOR
THE SLOWING DOWN OF FAST NEUTRONS

by

Thomas Martin Mills

Thesis Advisor: T. J. Williamson

June 1971

# Space Dependent Model for
# the Slowing Down of Fast Neutrons

by

Thomas Martin Mills
Major, United States Army
B.S., Polytechnic Institute of Brooklyn, 1962

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
June 1971

## ABSTRACT

The slowing down of fast neutrons was analyzed by a multi-group method of discrete time and energy states coupled with a spatial harmonic expansion method to determine the neutron density in a homogeneous, isotropically scattering slab. Five neutron source geometries were studied for both a fissioning and a non-fissioning system.

Numerical results were obtained for the neutron flux, mean neutron energy and the neutron spectra for the one dimensional system using a harmonic mode expansion of up to six terms to determine the time-energy-space dependence.

TABLE OF CONTENTS

## LIST OF FIGURES

FIGURE

FIGURE

# ACKNOWLEDGEMENTS

I wish to thank Professor Theodore J. Williamson for his guidance, support and assistance in the completion of this work.

# I. INTRODUCTION

Existing and future experimental work in the study of pulsed fast-neutron assemblies require the ability to determine the time dependent solution of the neutron transport equation accurately in the time period extending from the nanosecond to the millisecond range. This work allows the experimenter to compare both measured response and the analytical predictions for the overall improvement of each. Safe operation and control of fast nuclear reactors demands a fast, accurate mathematical simulation.

Current research on the response of fast assemblies to pulsed neutron sources considers that the time dependence can be represented by an exponential decay of the form:

$$T(t) = \sum_i A_i \exp(-\lambda_i t)$$

where the constants $A_i$ and $\lambda_i$ can be related to the composition and geometry of the system. A common assumption is that all terms above the first or fundamental will decay very quickly and that all measurable quantities can be described by a single fundamental or pseudo-fundamental value for $\lambda_i$. Experimental effort is being directed to answering the question of whether or not such a pseudomode does in fact exist, and if not, to understand why not.

This work will investigate several of the aspects of this problem.

In their movement through a material, neutrons undergo interactions, scattering or absorption, with the atoms and suffer changes in direction and velocity as the result of both elastic and inelastic collisions. While this is a random process for an individual neutron, the overall process for considering the time history and motion of the bulk of the neutrons is ameniable to a statistical analysis. The Boltzman time-dependent transport equation:

$$\frac{\partial N}{\partial t}(\underline{r},v,\underline{\Omega},t) = -v\underline{\Omega}\cdot\nabla N(\underline{r},v,\underline{\Omega},t) - v\Sigma_t(v)N(\underline{r},v,\underline{\Omega},t)$$

$$+ \int_{\underline{\Omega}} \int_{v} v'\Sigma(v')N(\underline{r},v',\underline{\Omega}',t)\cdot$$

$$f(v',\underline{\Omega}'\rightarrow v,\Omega)d\underline{\Omega}'dv' + S(\underline{r},v,\underline{\Omega},t)$$

can describe the net processes occurring in any elemental volume of a physical system to account for all changes in the density of neutrons, subject to the usual conventions:

| | |
|---|---|
| $\underline{r}$ | Position of the volume element dV. |
| v | Speed. |
| $\underline{\Omega}$ | Direction. (The vector velocity would be defined as $\underline{v} = v\underline{\Omega}$). |
| $N(\underline{r},v',\underline{\Omega}',t)$ | Density of neutrons at time t having velocity $v'\underline{\Omega}'$. |
| $\Sigma(v')$ | Speed dependent cross section for neutrons with speed (v') |
| $f(v',\underline{\Omega}'\rightarrow v,\Omega)$ | Probability that a neutron having velocity ($v'\underline{\Omega}'$) will interact to become one velocity ($v\underline{\Omega}$). |

The neutron density variation with time considers:

(a) leakage of neutrons:

$$-v\underline{\Omega} \cdot \nabla N(\underline{r},v,\underline{\Omega},t),$$

(b) losses due to interactions (scattering or absorption) in the volume element:

$$-v\Sigma_t(v)\ N(\underline{r},v,\underline{\Omega},t),$$

(c) scattering of neutrons into the element phase space (position-velocity-direction) due to interaction of neutrons having other initial parameters:

$$\iint v'\Sigma_t(v')\quad N(\underline{r},v',\underline{\Omega}',t)\ C_t'f(v',\underline{\Omega}'-v,\underline{\Omega})\ dV'd\underline{\Omega}'$$

neutrons introduced into the element of phase space from a source in that element:

$$S(\underline{r},v,\underline{\Omega},t).$$

Complete derivations of the neutron transport equation can be found in Tait [1] and Davison [2] and is discussed in detail in many other texts on nuclear reactor physics.

During the past twenty years, numerous methods and approximations have been formulated to solve this intergro-differential equation, either in a closed analytical form, or by a numerical approximation, for both the time dependent and time-independent cases. While a major portion of this effort has been in obtaining the solutions applicable to thermal reactors, uranium-235 systems, the need to conserve

fuels into the distant future requires fast breeder reactors. Adequate understanding of the complete time, space and energy dependence of fast neutron populations requires the development of new analytic and simulation models for the fast neutron transport and fission problem.

Analytical solutions are often based on extension of the existing models for thermal systems. These solutions usually assume 1/v or constant neutron cross sections, with the age-diffusion theory [3] considering the steady state, time independent cases and with perturbation analysis for time dependent solutions [4]. These provide representations of the steady state systems, and simple kinetic models for fast reactors.

(1) Numerical methods generally deal with the time independent solutions of the transport equation by two principal methods:

Multi-group theory accounts for all energy dependent parameters by dividing the range of neutron energies into some finite number of intervals and defining average values of the energy dependent terms for each energy group. This method is frequently used to study the energy and time relations and the energy and position dependence [5]. This method has been used to consider up to several hundred groups between fission energies and the thermal condition [6].

(2) Monte Carlo is a statistical method to consider the interactions of individual neutrons to determine principal integral or system values for several spatial regions using multiple energy groups. The study of the individual

neutron "life history" from its birth to its eventual loss, allows determination of such parameters as the effective multiplication factor ($K_{eff}$). Computational requirements for this method limit its usefulness mainly to time independent calculations, and to a limited extent [7], the time dependent problem using very few velocity groups.

The description of the slowing down and thermalization of neutrons as a probabilistic Markov chain process was proposed by Perkel [8] in 1960, and expanded to consider the thermal energy region (0.00002 to 1.034 eV) by Ohanian and Daitch [9] for the study of the time dependent neutron spectra. Jenkins and Daitch [10] have extended this technique to formulations of the study of a pseudo-fundamental mode decay for time-energy response of pulsed fast systems, assuming that the system response can be represented by the lowest Fourier spatial mode of the time dependent diffusion equation for two energy regions: .001 to 1.0 eV and 1.0 to 1000 Kev.

Williamson and Albrecht [11] have extended the Markov chain method of treatment for the slowing down problem to cover the continuous range from 10.5 MeV to thermal energies.

Menzel, *et al,* [12] have employed the method of Ohanian and Daitch to provide a complete harmonic expansion method for the study of the space-time-energy response of pulsed systems in the interval 0.01 to 1.0 eV.

The purpose of this thesis is to apply the numerical methods of the Markov chain process to describe the energy

and time response of pulsed fast systems, with the harmonic expansion techniques for the spatial dependence similar to Menzel, in the interval 10.5 Mev to thermal energies.

Current work on the pulsed neutron response of fast assemblies, considers that the time decay can be represented by some exponential function of the form:

$$T(t) = \sum_{i} A_i \exp(-\lambda_i t)$$

where the constants $\lambda_i$ and $A_i$ can be related to the composition and geometry of the assembly.

In this study, the fundamental and higher harmonic space modes are combined with energy-time expressions to determine the effectiveness of this technique is simulating the experiments conducted in the laboratory. Experiments suggest that the decay of the neutron population is often <u>not</u> best described by the pseudo-fundamental mode. Hopefully, this work will aid in determining if this is due to the effects of a constantly varying velocity spectrum, or to the persistance of higher spatial modes of the time response remaining for longer periods of time.

## II. THE SLOWING DOWN EQUATION

### A. THE BOLTZMAN TRANSPORT EQUATION

Starting with the general form of the time dependent transport equation, reasonable assumptions which will permit this intergro-differential equation to be restated as the slowing down equation are:

(1) The neutron density $N(\underline{r},v,\underline{\Omega},t)$ is isotropically distributed in $\underline{\Omega}$ at all points in the physical media.

(2) All sources and scattering kernels are isotropic.

(3) No delayed neutron sources.

(4) The physical media is composed of locally homogeneous isotropic materials.

(5) Spatial dependence is included within the bounds of the diffusion approximation.

(6) Boundry conditions are that the neutron density is zero at the extrapolated boundry and that the extrapolation distance is constant for all energies.

Conditions 1, 2 and 4 permit the equivalence of the expression for the local streaming or leakage of neutrons from a volume element to be described as:

$$-v\underline{\Omega}\cdot\nabla N(\underline{r},v,\underline{\Omega},t) = D(v)\nabla^2 N(\underline{r},v,\underline{\Omega},t) = -B^2 D(v)\ N(\underline{r},v,\underline{\Omega},t)$$

where $B^2$ is called the geometric buckling.

Within this analysis, a source that is a delta function in time will be considered as the mathematical equation of

the pulsed neutron source with a very small time duration, which can be written as:

$$S(\underline{r},v,\underline{\Omega},t) = S'(\underline{r},v,\delta(t))/4\pi.$$

Adjusting all constant terms to account for the isotropic conditions of the geometry under study, and to match the primary condition that the neutron density at a time (t=0) will be equivalent to the value of the source strength at that time we have:

$$N(\underline{r},v,0) = S(\underline{r},v,0) = \sum_n R_n(\underline{r})\ F_n(v,0),$$

where solutions of the form $N(\underline{r},v,t)$ serve as Green's functions for the pulsed conditions with an arbitrary time and spatial parameter.

Starting with the basic assumption that the neutron density at any time can be described as an infinite sum of solutions of the form:

$$N(\underline{r},v,t) = \sum_n R_n(\underline{r})\ F_n(v,t),$$

it is possible to transform each of the terms of the general transport equation to the slowing down equation in the following manner:

$$(1)\quad \frac{\partial N(\underline{r},v,\Omega,t)}{\partial t} = \sum_n R_n(\underline{r})\ \frac{\partial F_n(v,t)}{\partial t}\ .$$

$$(2)\quad -v\underline{\Omega}\cdot\nabla N(\underline{r},v,\underline{\Omega},t) = -D(v)\nabla^2 N(\underline{r},v,t)$$

$$= -D(v)\sum_n \nabla^2 R_n(\underline{r})\; F_n(v,t).$$

$$(3)\quad \int v'\Sigma_t(v')\int N(\underline{r},v',\underline{\Omega}',t)C_{t'}f(v',\underline{\Omega}'\rightarrow v,\Omega)dv'd\underline{\Omega}' =$$

$$\int v'\Sigma_t(v'\rightarrow v)\sum_n R_n(\underline{r})\; F(v',t)dv'$$

which reduces to the now modified form of the slowing down equation:

$$\sum_n R_n(\underline{r})\left\{\frac{\partial}{\partial t} F_n(v,t) + \Sigma_t(v)\; F_n(v,t) - \int v'\Sigma_t(v'\rightarrow v)\cdot F_n(v',t)dv'\right\} + \sum_n D(v)\; F_n(v,t)\nabla^2 R_n(\underline{r}) =$$

$$S'(\underline{r},v,t)$$

substituting for the spatial-velocity dependent delta function source, the eigenfunctional expression of the slowing down equation is obtained:

$$\sum_n\left\{R_n(\underline{r})\left\{\frac{\partial}{\partial t} F_n(v,t) + \Sigma_t(v)\; F_n(v,t) + \int v'\Sigma_t(v'\rightarrow v)\cdot F_n(v',t)dv'\right\} + D(v)\; F_n(v,t)\nabla^2 R_n(\underline{r})\right\} = \sum_n R_n(\underline{r})\; F_n(v,0).$$

Consolidating all expansion functions of the index-n in the single summation,

$$\sum_n \left[ R_n(\underline{r}) \left\{ \frac{\partial F_n(v,t)}{\partial t} + \Sigma_t(v)\; F_n(v,t) + \int v'\Sigma_t(v'\; v) \cdot \right.\right.$$

$$\left.\left. F_n(v',t)dv' - F_n(v,0) \right\} + D(v)\; F_n(v,t)\nabla^2 R_n(\underline{r}) \right] = 0$$

it is now possible to separate the spatial dependence from the energy time relations via the harmonic buckling factor, $B_n^2$, to two expressions:

$$\frac{\nabla^2 R_n(\underline{r})}{R_n(\underline{r})} = +\; B_n^2$$

and:

$$-D(v)B_n^2\; F_n(v,t) = \frac{\partial}{\partial t}\; F_n(v,t) + \Sigma_t(v)\; F_n(v,t)$$

$$+ \int v'\Sigma_s(v'-v)\; F_n(v,t)dvdt - F_n(v,0).$$

Final clearing and rearrangement of all terms yield the decoupled expressions:

$$\nabla^2 R_n(\underline{r}) - B_n^2\; R_n(\underline{r}) = 0$$

$$\frac{\partial}{\partial t}\; F_n(v,t) + B_n^2\; D(v)\; F_n(v,t) + \Sigma_t(v)\; F_n(v,t)$$

$$+ \int v'\Sigma_s(v' \rightarrow v)\; F_n(v',t)dv' = 0$$

with $R_n(\underline{r})$ a time independent function and $F_n(v,t)$ a space independent function.

## B. MOD-5: THE DISCRETE STATE APPROACH

In the separated form of the slowing down equation, the velocity-time harmonic term must satisfy the same form of the intergro-differential equation as the space independent neutron density function.

$$\frac{\partial}{\partial t} N(v,t) = -vD(v)\ B^2N(v,t)\ -v\Sigma_t(v)N(v,t) + \int v'\Sigma_s(v'\to v)\ N(v',T)\ dv' + S(v,t).$$

Any method that provides a solution to the space independent slowing down equation can be applied to solve the velocity-time harmonic equation.

Williamson and Albrecht [11,13] have developed a stochastic model for neutron moderation that provides a numerical solution of the slowing down of fast neutrons in a finite media. Since the slowing down process is a continuous time process in which collisions may occur at any time, having a continuous range of energy transition possibilities, it may be classified as a continuous time, continuous state Markov process. If the neutron cross sections remain constant with time, the slowing down may be classed as a stationary Markov system.

Application of the formalism of the Markov process calculations provides the capability to perform straightforward computer solutions for a discrete time-velocity model of the space independent equation. The probability that a neutron will be in a finite width velocity state, $v_i$, at a time, $t_j$,

is calculated by a computer program, MOD-5, which follows the slowing down of neutrons by determining a probability density parameter $F(v_i, t_j)$ which is the probability that a neutron will be in the velocity state $v_i$ at the discrete time $t_j = n\Delta t$.

The Markov process describes the variation of a probability density vector, $s(n\Delta t)$, which describes the probability that a neutron will be located at some state in the system during its entire lifetime from birth to death. This state vector is given as:

$$\bar{s}(n\Delta t) = s_1, s_2, s_3, s_4, \cdots , s_N)$$

where $s_i$ is the probability that the neutron is in the state bounded be velocities $v_i$ and $v_{i-1}$ (where $v_i < v_{i-1}$) at the time $n\Delta t$. This method presumes that the initial state vector, neutron source, $\bar{s}(0)$, can be described for time $t=0$.

The problem description necessary for application of a MOD-5 type of approximation are:

(a) The velocity range of interest is divided into "M" discrete states.

(b) Collision physics is applied to construct an array, $P(i,j)$, of the discrete Markov transition probabilities of a neutron going from velocity state-i to state-j in the discrete time period $\Delta t$.

While the details of the Markov transition matrix are discussed in several sources [14], the main considerations are that the array for the slowing down approximation is

upper triangular with only positive elements, with the elements of a given row summing to unity, which states that all possible outcomes of the process are considered.

If the initial Markov transition matrix describes the probability for a transition within a specified time interval, $\Delta t$, the probabilities for a transition at a time $\{n\Delta t\}$ can be determined as:

$$\bar{\bar{P}}_{n\Delta t} = (\bar{\bar{P}}_{\Delta t})^n.$$

The probability of a neutron being in any state at a time $(-n\Delta t)$ can be determined from the (n-1)th step in the matrix product as:

$$\bar{s}(n\Delta t) = \bar{s}((n-1)\Delta t) \cdot \bar{\bar{P}}\Delta t.$$

Both the program MOD-5 and the harmonic expansion method employ several series of notations and can result in some conflict with existing literature. In this work, the following definitions and terms will be used:

| | |
|---|---|
| $\bar{s}(n\Delta t)$ | the probability density vector of a neutron being in any state in the system at the time $n\Delta t$. |
| $S_{i,j}$ | the probability that a neutron will be in a state "i" at a time $t_j$, |

where the time and total density vector can be written as:

$$t_j = n\Delta t$$

$$\bar{s}(t_j) = (S_{1,j}, S_{2,j}, \cdots\cdots\cdots, S_{I,j}).$$

While this form of notation applies to the case of a single decay mode, the expansion to a multiple harmonic expansion method requires the addition of another indexing parameter-n to indicate the particular harmonic mode. This results in the probability density expressions being written as:

$s_{n,i,j}$ - probability of the neutron fraction of the n-th harmonic mode being in a velocity state $v_i$ at the time $t_j$.

$F_n(v_i,t_j)$ - n-th harmonic term of the time velocity dependence of the neutron probability density.

The contribution of each harmonic term of the neutron density is the product of the spatial component and the probability density component:

$$N_n(x,v_i,t_j) = R_n(x)\ F_n(v_i,t_j)$$

where the space component, $R_n(x)$, is a measure of the fractional contribution to the neutron population of the n-th harmonic mode, and the term, $F_n(v_i,t_j)$ is the probability of a neutron being in the velocity state $v_i$ at time $t_j$ for the n-th mode of the time decay harmonic.

In the following discussions, the equivalence of the terms is described as:

$$S_{n,i,j} = F_n(v_i,t_j)\Delta v_i$$

where $\Delta v_i$ is the width of the velocity state about the value $v_i$.

# III. GEOMETRICAL MODELS

## A. SOURCE FUNCTIONS

The main objective of this thesis is the investigation of the harmonic mode expansion method for the space, time and energy dependence of the neutron density in a physical system. Simple source functions representative of actual systems are desired to demonstrate this method. Three categories of source type were considered: a finite volume source and a first collision source as interior sources in the system, and a wide beam exterior source condition.

The test geometries are:

(a) Finite Volume Source - a finite volume element, located on an axis of symmetry and at an off axis position is considered to represent a hole or cavity in the slab. This might be envisioned as a beam port or localized neutron generator capable of providing a pulse of monoenergetic neutrons in a small volume in the slab. This is represented as a fourier square pulse on the x-axis of the system.

(b) First Collision Source - this exponential distribution of neutrons in the system is considered as a more realistic approach for the response of a physical system than the fixed source in volume condition. Rather than assume that the initial burst of neutrons would remain in the fixed volume element, it can be argued that they would initially travel throughout the system with an exponential form.

As with the finite volume type source, this source type is studied for both the symmetric and unsymmetric positions.

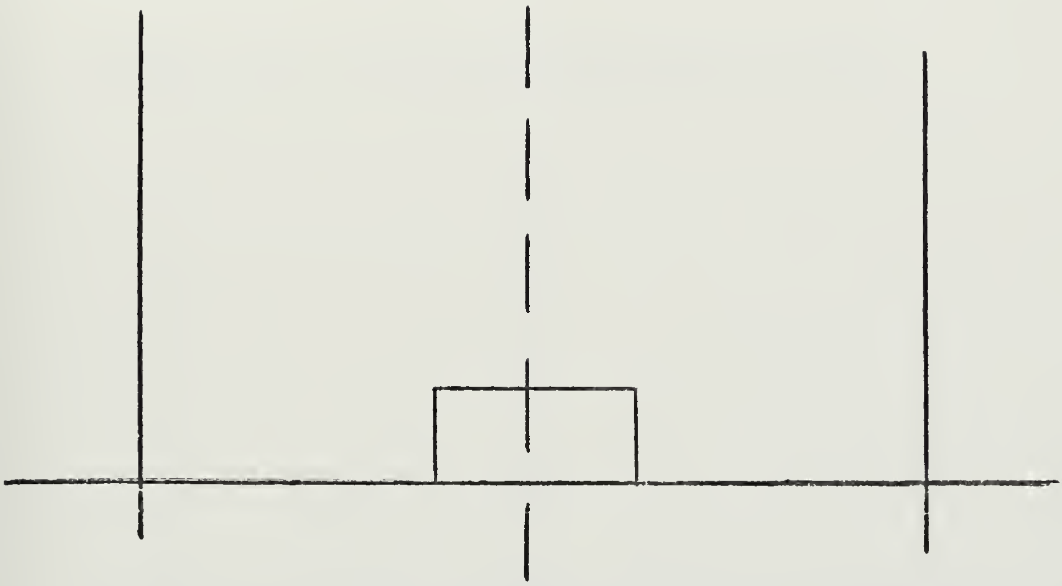
(c) Exterior Source Beam - a higher level of complexity in the source geometry selection process is the physical situation where the assembly under study is subject to a wide beam of fast neutrons from an external point. This can be considered as a very simple approximation to a first collision response from a plane source of neutrons on the surface of the slab. While this may be considered to follow an exponential type distribution through the slab, it is approximated as a straight line or ramp function within the confines of the slab.

## B. CHOICE OF DIMENSIONALITY

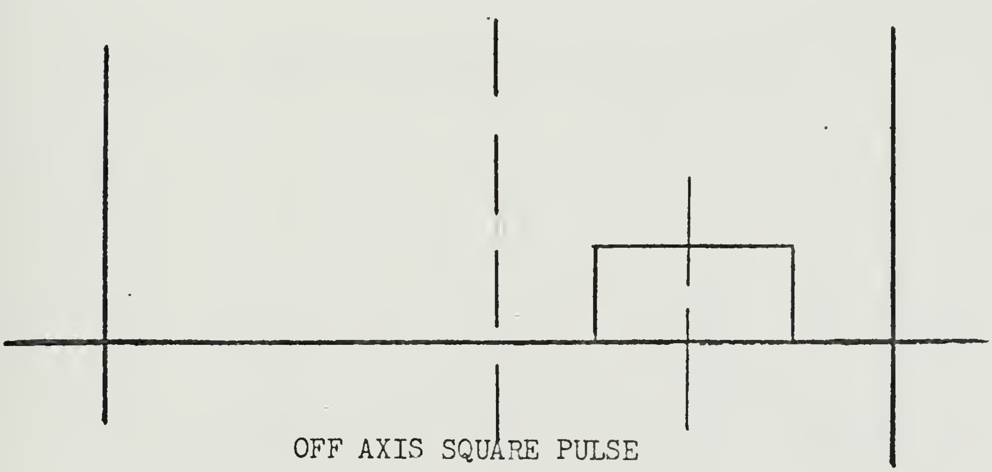
The principal goal of this work is the development of a useful mathematical model to determine the spatial dependent response of a physical system to a burst of fast neutrons. In this determination, it will be necessary to refer to the neutron density in terms of three main parameters - time, space and energy. The derivation of all equations and principal relationships will be written for the time, position and velocity of the neutrons. Equivalent expressions can easily be obtained using either the neutron energy (E) or lethargy (u), using the relationships:

$$E = \frac{1}{2} m_n v^2$$

$$u = -\ln(E_0/E) = -2 \ln(v_0/v).$$



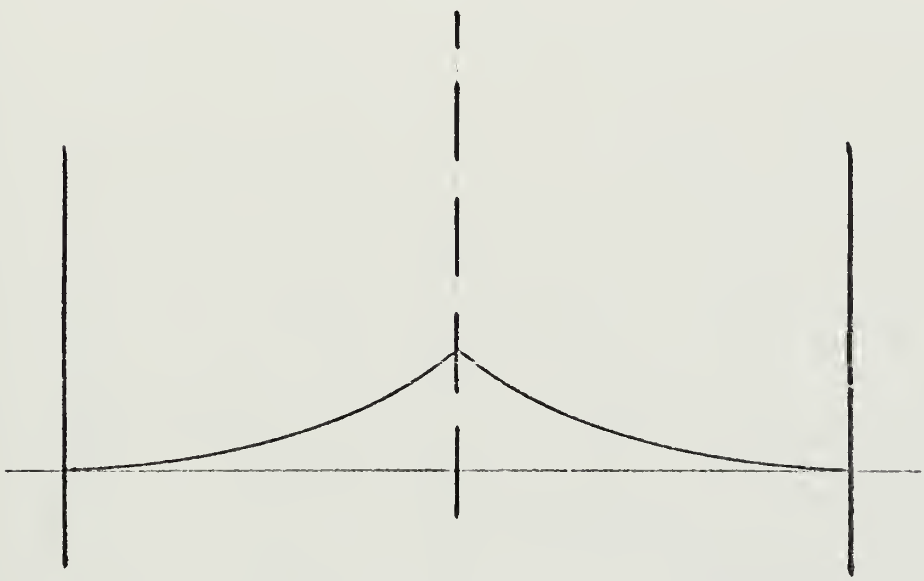
CENTRAL AXIS SQUARE PULSE



OFF AXIS SQUARE PULSE

FIGURE 1 SOURCE GEOMETRY SQUARE PULSE

CENTRAL AXIS FIRST COLLISION-EXPONENTIAL SOURCE



OFF AXIS FIRST COLLISION-EXPONENTIAL SOURCE

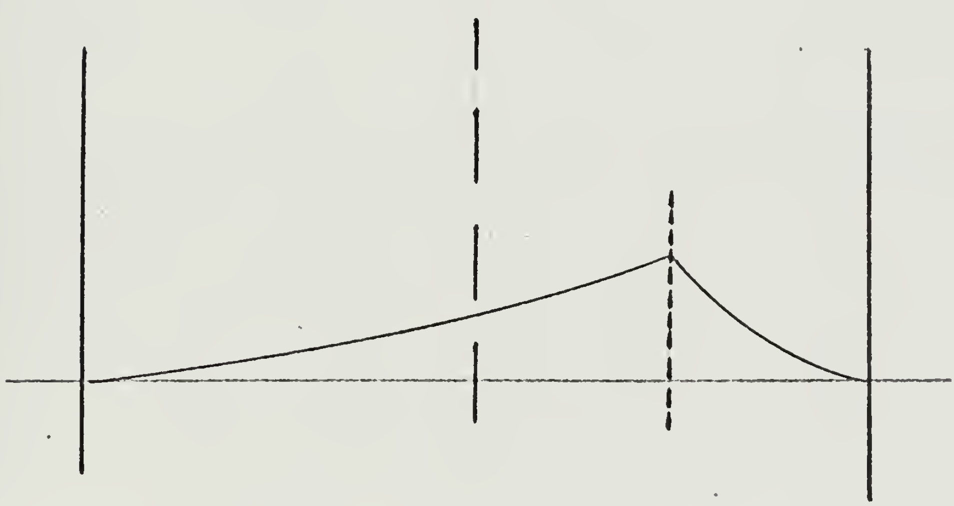


FIGURE 2    SOURCE GEOMETRY FIRST COLLISION SOURCE

FIGURE 3 SOURCE GEOMETRY EXTERIOR SOURCE RAMP FUNCTION

Employing the appropriate expressions, it can be easily shown that equivalent terms for the neutron density, $N(\underline{r},v,t)$, the time-space-velocity form can also be modified to give the time-space-energy or time-space-lethargy form. While the velocity dependent form is used for the purposes of demonstration within this paper, all calculations performed by the programs MOD-5 and MIL-6 employ the lethargy dependent forms.

In the determination of numerical solutions for the slowing down equation, reasonable methods and simple systems are desired to demonstrate the harmonic expansion method for the time, velocity and spatial expressions, where the method is general enough to be easily revised to handle more difficult and complex problems. As shown previously, the neutron density can be determined by the summation of a harmonic series whose terms are the product of a space function and a time-velocity function.

$$N(\underline{r},v,t) = \sum_n R_n(\underline{r})\, F_n(v,t) \quad .$$

where the spatial dependent function, $R(\underline{r})$, satisfies the differential equation:

$$\nabla^2 R_n(\underline{r}) - B_n^2 R(\underline{r}) = 0.$$

In order to retain the flexibility to study and compare the response of physical systems to various source types, the selection of the dimensionality for problem selection posed some difficult choices.

The spatial functions must form an orthogonal set, a condition easily met by the Fourier expansion methods for the one dimensional problem using trigonometric functions for the cartesian geometry, but requires the use of regular Bessel functions for cylindrical geometry, and the spherical Bessel and spherical harmonic functions for the spherical geometry. A practical consideration was to obtain a balance demonstrating the harmonic expansion application with reasonable computing requirements. The more complex the spatial functions, the longer computing time and large core storage space required. The trigonometric functions can be determined quickly, while the existing methods available for the Bessel function calculations require approximately three times as much computing time.

In all three of the principal geometrical coordinate systems (cartesian, cylindrical and spherical), studies of sources located at an axis of symmetry resolve themselves into some form of the one dimensional problem. However, the study of sources located off the axis of symmetry of the system, while seeming reasonable conditions to analyze, can lead to some very messy mathematics, such that the utility of the method can become lost in the details. While these problems can be considered, the one dimensional case would allow easy comparison of all symmetrical and nonsymmetrical cases without initially confusing the reader or user of this technique prior to demonstrating the method. After accomplishing that aim, the multi-dimensional cases (two or three dimensions) can be investigated.

A practical matter in the consideration of the harmonic expansion method involves the retention of data within the computing machinery while running the basic problem or one of the many variations that can be considered by this method.

An important consideration while developing the computer programming for these calculations involved the selection of methods to transfer data between the two computational routines. The program MIL-6 was written to process data input from punch cards, nine-track magnetic tape, or the IBM-2314 series of magnetic disk. The program MOD-5 was run sequentially to determine the velocity and time characteristics of the physical system for each harmonic mode of the buckling constant ($B_n^2$). This generally requires one minute of computing time for the IBM-360/67 with 70 velocity states, requiring 200k bytes of core storage for each harmonic mode. Thus, a six-mode expansion would require approximately six minutes of central processor time.

Assuming that the problem under study is going to deal with 150 discrete velocity states, 20 time step values and a minimum of six harmonic modes of the functional data, $F_n(v,t)$, as produced by program - MOD-5, one is faced with the retention of 18000 numbers which are required either in core storage or in an on-line direct access device. In the IBM-360/67, all real numbers require a 4 byte location, and a minimum problem would require 72k bytes (12 per mode)

available during the running of the problem. Once the primary calculation of the density vector $N(\underline{r},v,t)$ is done, data retention is required for the total system $N(\underline{r},v,t)$ which would be estimated as 4.0 (position points) (dimensions) (energy states) (time intervals) which can easily become a very large number when considering 40 or 50 points in a physical system, for cases in two or three dimensions.

The crux of the matter is that the one dimensional model will allow the utility of this approach to be demonstrated easily with relatively simple, easily compared Fourier series approximations in sine and cosine terms, rather than the functional forms required for solution in three dimensional cylindrical or spherical coordinate systems for sources not located at the principal axis symmetry.

## C. SOURCE FUNCTION APPROXIMATIONS

Within the bounds of the one dimensional spatial model, the five source functions can be described by comparatively simple Fourier series approximations:

$$S(x,v_i,0) = \sum_n (A_n \cos(k_n x) + C_n \sin(k_n x))$$

for a representation of the source geometry within the confines of the slab.

The following labeling conventions are used in defining the terms of the Fourier series representation:

$S_o$ source strength-total neutrons in the pulse.

$x_o$ half width of the slab.

$x_1$ half width of the finite volume source.

$x_2$ mid-point of the source for the off axis conditions

L diffusion length for the source velocity neutrons.

$\lambda$ the extrapolation distance for the source neutrons.

The source function approximations are subject to the conditions:

Square Pulse

On-Axis
$$S(x,v_j,0) = \begin{cases} 0 & x < -x_1 \\ S_o & -x_1 \leq x \leq x_1 \\ 0 & x > x_1 \end{cases}$$

Off-Axis
$$S(x,v_j,0) = \begin{cases} 0 & x < x_2 - x_1 \\ S_o & x_2 + x_1 \leq x \leq x_2 - x_1 \\ 0 & x > x_2 + x_1 \end{cases}$$

First Collision Source

On-Axis
$$S(x,v_j,0) = \begin{cases} S_o\ e^{+ax} & x \leq 0 \\ S_o\ e^{-ax} & x \geq 0 \\ 0\ - & x > x_1 \end{cases}$$

Off-Axis
$$S(x,v_j,0) = \begin{cases} S_o\ e^{-a|x-x_2|} & -x_o \leq x \leq x_o \\ 0 & x < -x_1 \end{cases}$$

Exterior Source
$$S(x,v_j,0) = \begin{cases} mx + b & -x_o \leq x \leq + x_o \\ m = \dfrac{S_o}{2x_o^2} & b = \dfrac{S_o}{2x_o} \end{cases}$$

| Source Type | Series Representation | Fourier Coefficients |
| --- | --- | --- |
| (1) Square Pulse on axis, n = positive odd integers | $A_n \mathrm{Cos} k_n x$ | $k_n = \frac{n\pi}{2(x_o+\lambda)}$; $A_n = \frac{S_o \sin(k_n x_1)}{k_n x_o}$ |
| (2) Square Pulse off axis, n = positive odd integers | $A_n \mathrm{Cos} k_n (x-x_2)$ | $k_n = \frac{n\pi}{2(x_o+x_1+\lambda)}$; $A_n = \frac{S_o \sin k_n x_1}{2n x_1}$ |
| (3) First Collision Source on axis, n = positive odd integers | $A_n \mathrm{Cos} k_n x$ | $k_n = \frac{n\pi}{2(x_o+\lambda)}$; $A_n = \frac{S_o (-1)^{\frac{n-1}{2}}}{n(\frac{1}{4}+(x_o/nL)^2)(1-\exp(-x_o/L))}$ |
| (4) First Collision off axis source, n = positive integers | $A_n \mathrm{Cos} k_n x + C_n \mathrm{Sin} k_n x$ | $k_n = \frac{n\pi}{2(x_o+x_2+\lambda)}$; $A_n = \frac{S_o(\mathrm{Cos} k_n x_1 - k_n L \mathrm{Sin} k_n x_1 - (-1)^{n+1} \sin(x_1/L))}{L(1+(k_n L)^2)(1-\exp(-(x_o+x_1)/L))}$ |

TABLE I. Fourier Expansion Coefficients.

TABLE I. - Continued

| | | |
|---|---|---|
| | | $C_n = \frac{S_o(\mathrm{Sin}k_n x_1 - k_n L\mathrm{Cos}k_n x_1 - (-1)^n \sinh(x_1/L)}{L(1+(k_n L)^2)(1-\exp(-(x_o+x_1)/L))}$ |
| (5) Exterior Ramp Function Source | $A_n \mathrm{Sin}k_n x$ | $k_n = \frac{n\pi}{4(x_o+\lambda)}$ |
| n = positive integers | | $A_n = \frac{(-1)^{n+1} 2\, S_o}{n\pi\, x_o}$ |

## D. ENERGY-VELOCITY-LETHARGY

The slowing down equation can be written in several forms to describe the variation of the neutron energy with time. For computational purposes, it is often convenient to describe the kinetic parameter in terms of the neutron velocity, while at other times, the kinetic energy of the neutron would provide a better description. There are three main terms that can be used interchangeably to describe the neutron during the slowing down process: velocity, energy and lethargy. Any expression in one of these parameters can be transformed to the appropriate form in the others by the relationships:

| | |
|---|---|
| Energy-velocity | $E = m_n v^2/2$ |
| Energy-lethargy | $-\ln(E/E_o) = u$ |
| Velocity-lethargy | $-2\ln(v/v_o) = u.$ |

By these expressions, the parameters, $E_o$ and $v_o$, describe the neutron reference or source condition with the added convenience that as the neutron slows down, its lethargy increases, while for the velocity and energy expressions, the slowing down process obviously results in the neutron going to a state with a smaller value.

Simple relations between the derivatives of these expressions will allow the slowing down equation to be easily revised with the desired terms:

$$dE = m_n v \, dv$$

$$du = \frac{-2\ dv}{v} = \frac{-dE}{E}\ .$$

In terms of a neutron density function, $N_1(v)$ and the differential relations, all three forms of the neutron density are:

$$N(v)\ dv = N(E)dE \quad N(v) = m_n vN(E) = (2m_n E)^{\frac{1}{2}}N(E)$$

$$N(E)\ dE = N(u)du \quad -E\ N(E) = N(u)$$

$$N(v)\ du = N(u)du \quad N(u) = \frac{-v}{2}\ N(v).$$

## IV. APPLICATIONS

### A. NEUTRON DENSITY

The neutron density can be determined in the numerical model by the infinite series expression:

$$N(\underline{r},v,t) = \sum_{1}^{N} R_n(\underline{r})\ F_n(v,t)$$

such that the series representation converges to the analytical value as N increases to infinity. In the numerical method for this calculation, the series representation will be determined as a finite series, truncated to the first 3 to 6 terms. In this form of the approximation, the error introduced is principally that of under estimating the true value if an even number of terms are used, while slightly over estimating with an odd number of terms.

The truncation error of the finite series approximation can be estimated from the initial conditions of the system at time equal zero as the neutron density at that time must equal the initial source strength:

$$N(\underline{r},v,0) = S(\underline{r},v,0).$$

If the total source strength ($S_o$) is obtained by integrating over all velocities and the source volume, the source strength can be given by the series approximation:

$$S_o = \int_v \int_r S(\underline{r},v,0)\ dv\ d\underline{r} = \int_v \int_r \sum_n R_n(\underline{r})\ F_n(v,0)\ dv\ d\underline{r}.$$

With the use of a truncated series approximation, the error introduced can be calculated as:

$$\text{Error} = S_o - \iint \sum_{1}^{N} R_n(\underline{r})\ F_n(v,0)\ d\underline{r}\ dv.$$

This normalization error is used to obtain a first order adjustment factor, $\Delta_0$, by the simple approximation:

$$\Delta_0 = \qquad \qquad \doteq \frac{\text{Error}}{S_o}$$

and a normalization adjustment is determined as:

$$\text{Source Normalization} = (1 + \Delta_0)\ S_o$$

such that the truncated series will give a numerical value equal to the initially defined parameter-$S_o$. This now gives the series approximation as:

$$S(\underline{r},v,0) = \sum_{1}^{N} (1 + \Delta_0)\ R_n(\underline{r})\ F_n(v,0)$$

where the artifice of the source strength normalization factor ($\Delta_0$) would be a succeedingly smaller correction as more terms are included in the finite series representation.

This adjustment of the numerical value of the source strength has no effect on the velocity-time functions, $F_n(v,t)$, as this is a constant multiplier included in the spatial dependent expressions.

In the discrete state, discrete time calculations of MOD-5, the probability that a neutron will be within a

specified velocity interval ($v_i$) at the finite time step ($t_j$), transforms the neutron density expression to the form:

$$N(\underline{r},v_i,t_j) = \sum_n R_n(\underline{r})\ F_n(v_i,t_j)$$

and the total neutron population at any fixed time can be found by integration over the spatial coordinate and summation over the velocity intervals:

$$N(t_j) = \int_r \sum_i \sum_n R_n(\underline{r})\ F_n(v_i,t_j)\ d\underline{r}.$$

B. NEUTRON FLUX

The neutron flux, a scaler quantity to describe the net flow of neutrons per unit time and unit area, is calculated as the product of the speed of the neutron and the number density of neutrons having that speed:

$$\phi(\underline{r},v,t) = v\ N(\underline{r},v,t)$$

and is approximated in this work as the discrete value:

$$\phi(\underline{r},v_i,t_j) = v_i\ N(\underline{r},v_i,t_j).$$

The total integrated flux can be analytically determined by an integration over all speeds, and in this work, is therefore approximated by a summation over all discrete states of the density parameter $N(\underline{r},v_i,t_j)$ to yield the result:

$$\phi(\underline{r},t_j) = \sum_i v_i N(\underline{r},v_i,t_j) = \sum_i \sum_n v_i R(\underline{r})\ F_n(v_i,t_j).$$

## C. MEAN ENERGY OF NEUTRONS

The spatial variation of the neutron mean energy provides valuable information on their migration and diffusion following the initial burst. This additional insight into the phenomena of "diffusion cooling" considers the general trend of the higher velocity components of the neutron population to leak out the boundries of the finite system. As such, the remaining neutrons, while still undergoing interactions which remove them or lower their velocity, have their total number decreased by this additional effect.

In the discrete velocity state, discrete time model, the mean energy of the neutrons at a point $\underline{r}$ in the slab is determined as:

$$\bar{E}(\underline{r},t_j) = \frac{\sum_i E(v_i)\, N(\underline{r},v_i,t_j)}{\sum_i N(\underline{r},v_i,t_j)} \; .$$

In comparing the results of this mean energy determination, two basic methods can be used: first, to study the variation in time of the mean energy at a fixed position, and secondly, to consider the ratio of the mean energy at all positions in the system, for all time periods, compared to a single reference position. Within this work, the mean energy is compared at each finite time period to the current value of the mean energy at the center of the slab, while the time variation of the mean energy at the reference point is followed in detail.

In the discrete time-velocity-space model, this is determined as:

$$\frac{\bar{E}(\underline{r},t_j)}{\bar{E}(\underline{r}_o,t_j)} = \frac{\dfrac{\sum_i E(v_i)\ N(\underline{r}',v_i,t_j)}{\sum_i N(r',v_i,t_j)}}{\dfrac{\sum_i E(v_i)\ N(r_o,v_i,t_j)}{\sum_i N(r_o,v_i,t_j)}}$$

where $\underline{r}_o$ is the reference point, x=0.0, and $\underline{r}'$ is a position within the slab.

## D. DETECTOR RESPONSE

Numerous methods [15] are currently utilized to determine the neutron spectra and flux in fast reactor test assemblies, for internal and external measurements. Fast neutron leakage is examined in time of flight tests to evaluate the time-varying spectra and neutron resonance absorption foils to study the internal spatial dependent spectra.

For this analytical model, no single existing device appeared to be compatable to cover the continuous time and energy response as calculated by the discrete state method. An analytical detector is therefore constructed that would respond to the time-velocity variations of the neutron density. For future applications of this model, a one dimensional version of an existing devices characteristics can be introduced.

The analytical detector is modeled as a proton recoil scintilation device capable of measuring the proton recoil of hydrogen-neutron collisions. The detector is considered to be filled with hydrogen at a molecular concentration as $H_2$ equivalent to an ideal gas at STP., with physical dimensions of a width 2 $X_4$ about a midpoint $\underline{r}_D=X_3$.

The detector response (DR) is determined using the ABN-26 group cross section [16] to be:

$$DR\ (\underline{r}_D, v_i, t_j) = \int \Sigma_s(v_i)\ v_i\ N(\underline{r}', v_i, t_j)\ d\underline{r}'$$

where the integral over the spatial coordinate $d\underline{r}'$ refers to the total detector volume.

The total response of the detector both for the total integrated flux over all velocities and over all times can be determined in this discrete velocity-time model by numerical integration and summation:

$$DRV(\underline{r}_D, t_j) = \sum_i DR(\underline{r}_D, v_i, t_j)$$

$$DRT(\underline{r}_D, t_j) = \sum_j DRV(\underline{r}_D, t_j)$$

where the final summation over all times from t=0 to $t=t_j$ provides the neutron fluence at the detector.

## E. MIL-SIX COMPUTER PROGRAM

A general purpose computer program, MIL-SIX, was written in FORTRAN IV to process the time-energy (lethargy) response

date produced by MOD-5 to determine within this version of the transport approximation, the neutron density function-N($\underline{r}$,u,t) for the one dimensional infinite slab geometry. Several minor modifications were made in MOD-5 to provide the ability to calculate the time-energy response of a system with multiple harmonic modes and to provide the neutron probability density for each lethargy state at the same time steps. The only form of output system tested involved the use of punch card output from MOD-5, however, provisions have been made to employ nine-track magnetic tape and/or on-line disk storage capabilities (IBM-2314 units). Program MIL-SIX provides the general user the ability to accept any of the three forms output from MOD-5, for up to six (6) harmonic modes of data. The program can be modified easily to adjust for a higher order approximation using more than six modes with a simple change in the size of the storage arrays in the program.

Program MIL-SIX is composed of the following subprograms and routines:

| Subroutine | Function |
|---|---|
| Main | Principal control portion of the program; handle all logical decisions on which routines the program will perform |
| INCON1 | Establishes all principal default parameters for the program; reads all problem definition statements (punch cards) which determine the parameters that the program |

will need to calculate the desired functions; provides the user with instructions and information concerning actions the program will perform for the stated problem run.

| | |
|---|---|
| READ10 | Processes all punch card output from program MOD-5. |
| READ11 | Processes all tape (9-track) from program MOD-5. |
| ESTIMATE1 | Provides guidance information of the execution time to be expected for the running of a particular problem. |
| MODEL 1 | Calculates the non-harmonic terms of the fourier space dependent functional expression for the source geometry specified for the particular problem; estimates the error introduced by the use of a finite number of terms in the fourier infinite series approximation; adjusts the values of the fourier non-harmonic coefficients to obtain a closer approximation to the infinite series representation; and provide a simple space dependent plot of the fourier expansion of the space dependent function on the on-line printer. |
| PHI1 | Principal function sub-program of the program: to calculate the fourier space dependent harmonic terms in the calculation |

of the terms of $R_n(x)$ (n=the harmonic mode, x=the space point) for the one dimensional model.

FOUT1($x_i$) Principal determination subprogram to calculate the series of values for the function - $N(x,u,t) = R_n(x_k)F_n(u_i,t_j)$ for summed response due to all harmonic mode to give the neutron density for the single point - $x_i$.

FOUT2 This subprogram will calculate the two principal space dependent relationships: $\phi(x_k,t_j)$ and $\bar{\bar{E}}(x_k,t_j$ the integrated (total) neutron flux and the mean energy of the neutrons at the point $(x_k)$ for a profile of 100 points across the slab for each time steps $(t_j)$, and will provide a graphical plot of the functional forms:

$$\phi(x_k,t_j) \text{ vs } x_k$$

$$\bar{E}(x_k,t_j) \ / \ \bar{E}(0,t_j) \text{ vs } x_k$$

via a on-line printing of a rough plot of the results or via the Cal-Comp Plotter for each set of discrete time step date $(t_j)$; also provides tabulated data on neutron flux, mean energy and mean enerby ratio.

FOUT3 Detector response: this routine considers the simple model of a hydrogen recoil detector located inside the slab. This is not a true device, but one which determines the function:

$$\phi(x_k, t_j) = \sum_u v(u)*\Sigma_t*\sum_n R_n(x_k)F_n(u_i, t_j)$$

using the integrated position dependent flux (calculated at 5 points within the assumed detector position); a simple average is performed to obtain the flux at the detector, where the energy dependent microscopic cross section ($\Sigma_t(u_i)$ for STP hydrogen is based on the ABN-26 group cross section tables. Starting with $\phi(x_k, t_j)$ - flux data, a numerical integration is performed to determine the integrated detector response. Output of the two functional forms of the detector response are given both by the on-line printer, and the cal-comp plotter.

FOUT4 Provides a simple spectral plot of the data input from program MOD-5, to show the distribution $F_n(u_i, t_j)$ for each harmonic mode versus the lethargy state, showing the movement of the neutron distribution to states of lower energy (higher

lethargy) with time for each mode of harmonic buckling.

FOUT5 Provides a more detailed spectral response plot of the MOD-5 input data, by consideration of the fourier harmonic coefficient $R_n(x_k)$ for the individual position point - $\underline{x}_k$ weighting the harmonic data - to give the response spectra: $N_n(x_k, u_i, t_j) = R_n(x_i) F_n(u_i, t_j)$ as a single plot for each harmonic mode of input data, and a total spectral response function $N(x_k, u_i, t_j) = \sum_n R_n(x_k) F_n(u_i, t_j)$, summed over all modes and plotted versus lethargy state - $u_k$ on the Cal-Comp plotter.

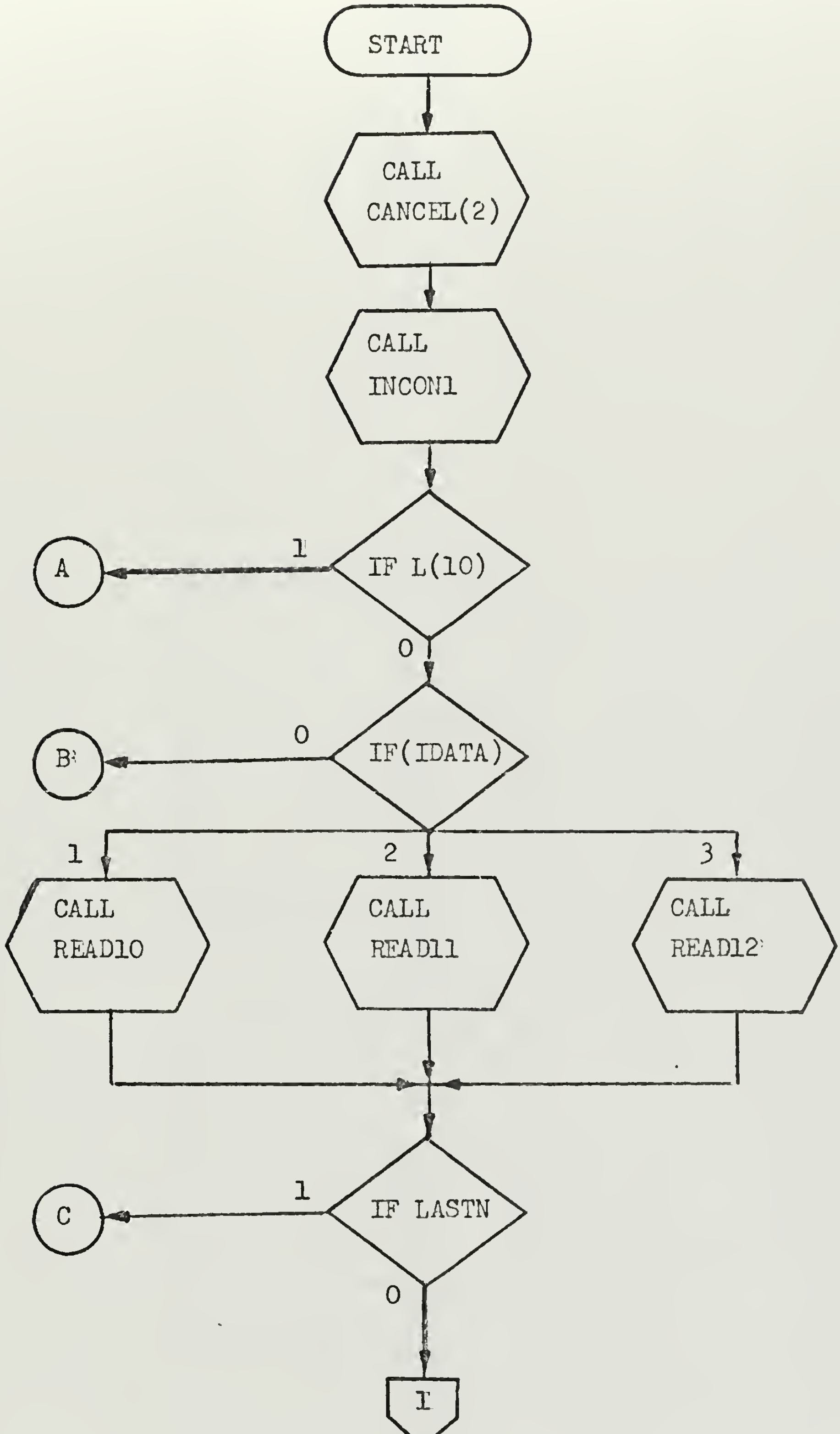
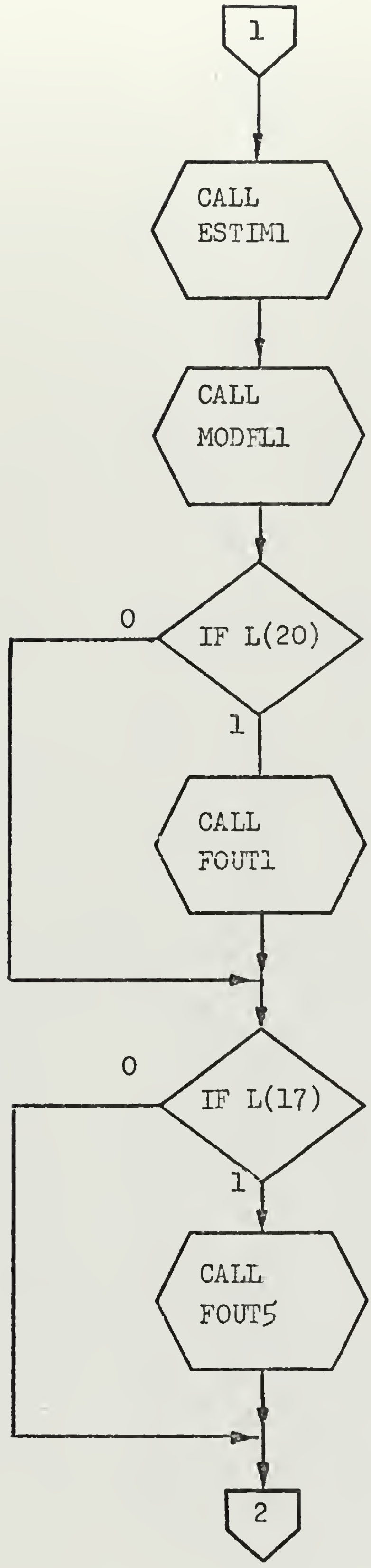
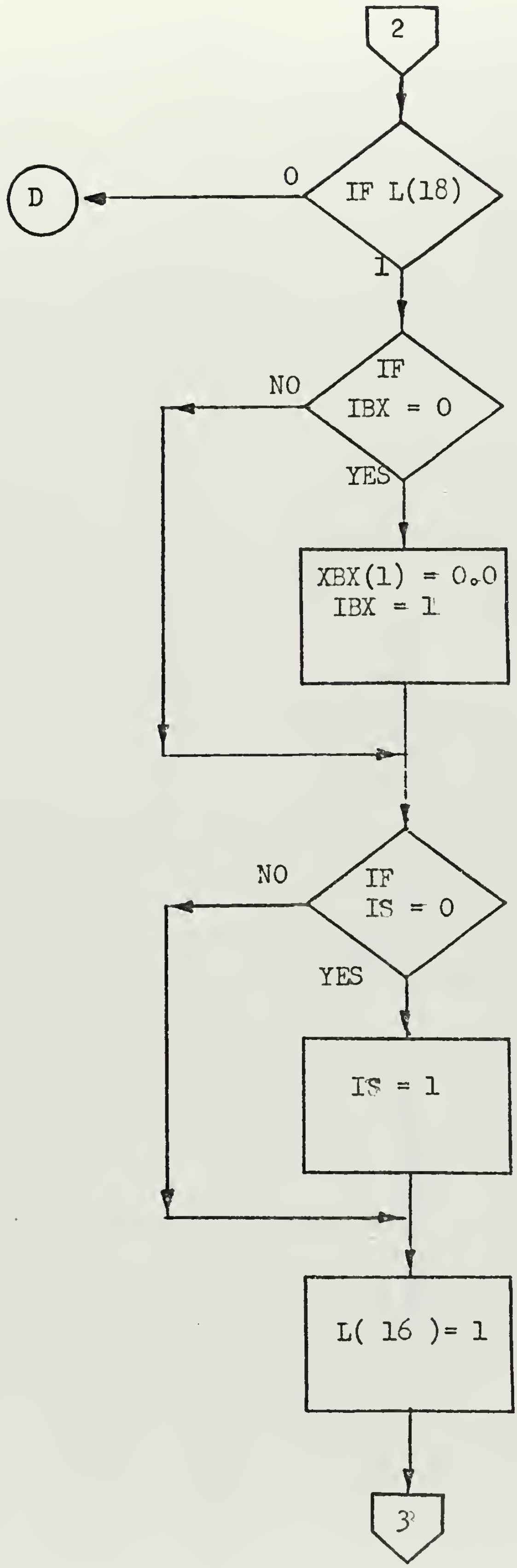
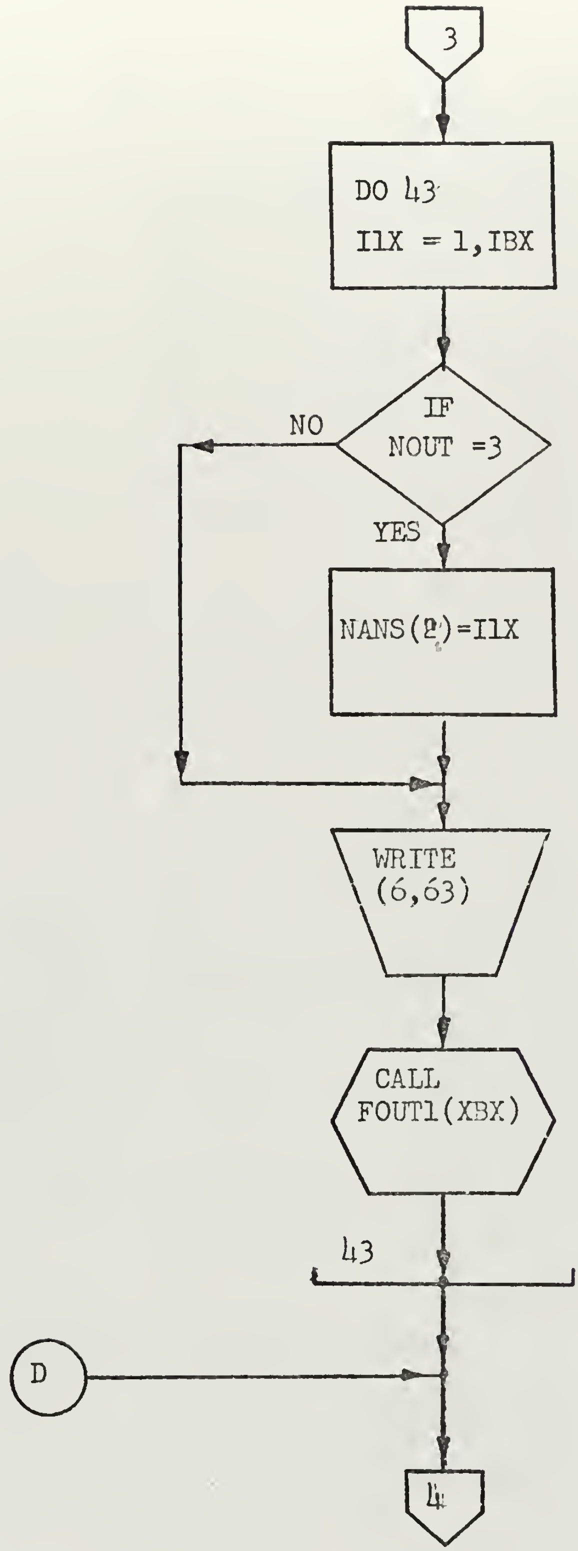
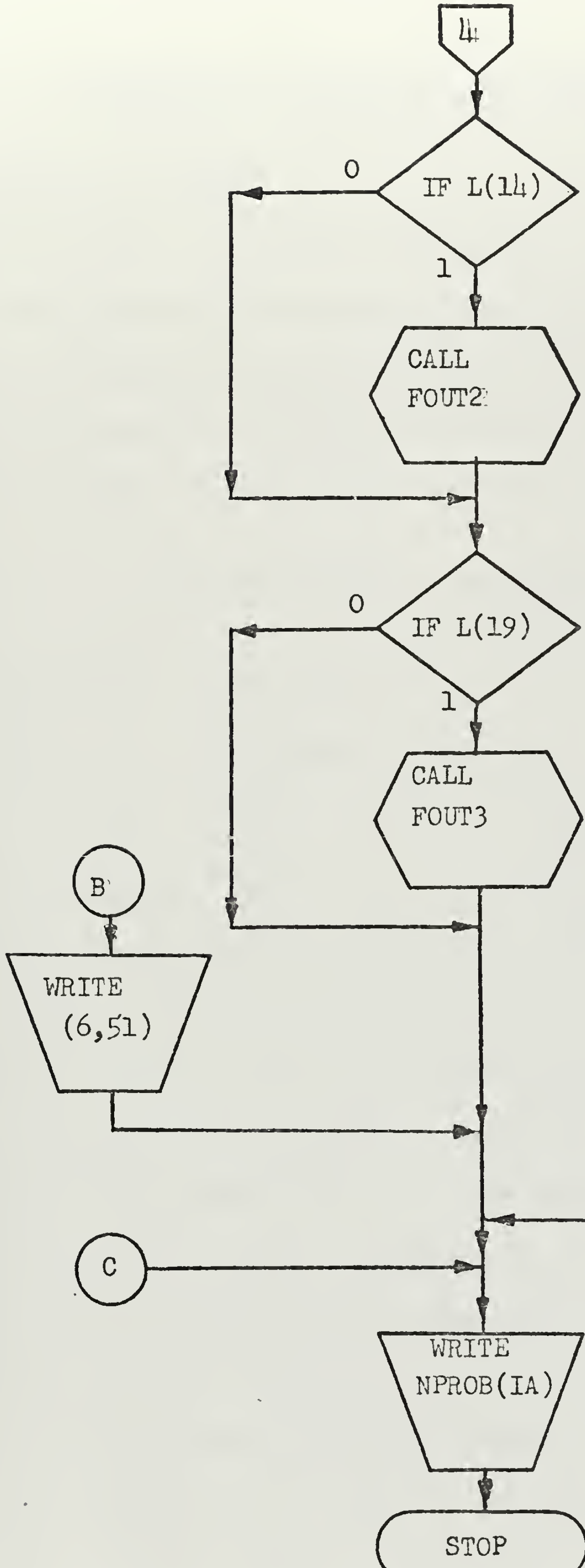



FIGURE 4 FLOW CHART COMPUTER PROGRAM MIL-SIX


1
CALL
ESTIM1
CALL
MODEL1
IF L(20)
0
1
CALL
FOUT1
IF L(17)
0
1
CALL
FOUT5
2


2
IF L(18)
0
D
1
IF
IBX = 0
NO
YES
XBX(1) = 0.0
IBX = 1
IF
IS = 0
NO
YES
IS = 1
L( 16 )= 1
3


3
DO 43
I1X = 1,IBX
IF
NOUT =3
NO
YES
NANS(2)=I1X
WRITE
(6,63)
CALL
FOUT1(XBX)
43
D
4


4
IF L(14)
0
1
CALL
FOUT2
IF L(19)
0
1
CALL
FOUT3
B
WRITE
(6,51)
A
WRITE
(6,53)
C
WRITE
NPROB(IA)
STOP

## V. DISCUSSION OF RESULTS AND CONCLUSIONS

### A. PHYSICAL SYSTEMS

The harmonic expansion method was examined for application to multiplying and non-multiplying systems, a beryllium slab and a modified form of the ZPR-3 assembly 6F. The physical parameters of the two systems are given in Table II. The nuclear characteristics were assumed to be:

| System | Constituents | Mass Density (gm/cc) | Total Modes |
|---|---|---|---|
| Beryllium Slab | Beryllium | 1.84 | 6 |
| ZPR-3 (6F) | Aluminum | 0.848 | 3 |
| | Iron* | 0.965 | |
| | U-235 | 2.622 | |
| | U-238 | 3.016 | |

(*Iron substituted for stainless steel.)

A pulse of one 2.46 MeV neutron was fixed as a delta function in time to study the decay of each harmonic component of the discrete state velocity-time response $F_n(v_i, t_j)$. Two versions of the slowing down process were examined:

(1) follow each harmonic mode to the same final time ($t_j$) as the fundamental mode,

(2) follow decay of each harmonic mode to the time step at which the probability density component for that mode was below 0.001 of the neutron remaining in the system.

Both the beryllium and the assembly 6F systems were studied to determine the variations in the neutron spectra

| | ZPR-3 (6F) | Beryllium Slab |
|---|---|---|
| Slab Half Width ($x_o$) | 17.50 | 25.00 |
| Source Half Width ($x_1$) | 2.00 | 2.50 |
| Diffusion Length | 9.79 | 7.085 |
| Mid Point or Off-Axis Source ($x_2$) | 12.50 | 12.50 |
| Detector Mid Point ($x_3$) | 10.00 | |
| Detector Half Width ($x_4$) | 1.00 | 12.50 |
| Fundamental | | 1.00 |
| Buckling Mode ($B_o{}^2$) | .00704 | .00093 |

All measurements in centimeters

TABLE II. Physical Parameters of Test Systems.

resulting from the inclusions of the higher harmonic modes. The persistance of the higher modes was observed in the early time history, but provided a negligible contribution toward the end of the decay time.

## B. DATA PRODUCTION

The neutron probability density data for the lethargy-time response was calculated by MOD-5 for all harmonic modes using two limits on the systems considered. MOD-5 normally selects those discrete time steps to provide the probability density data under its internal selection criteria as though all problems were a fundamental mode type calculation. This freedom in selecting the output time steps was over ridden by modification of the existing control sequency in the basic MOD-5 program.

The fundamental mode calculation was done for a system to obtain the fixed time intervals and output time steps for the higher harmonic mode calculations. With the "locking in" of the data output time steps, the higher harmonic modes were run to the same final discrete time as the fundamental mode. Unfortunately, as the higher modes decay more rapidly, this resulted in the abnormal termination of several computer runs when the probability density dropped to values smaller than the IBM-360/67 is capable of processing, i.e., below $10^{-75}$. As a result, all final computer runs were set to terminate when the probability density component for a mode dropped below an arbitrary limit of 0.001.

The time decay of the harmonic components is mainly affected by the leakage of neutrons from the system which is dependent on the buckling factor, $DB_n^2$, which appears in the time dependent diffusion equation as a term: $vD(v)B^2N(x,v,t)$.

A summary of the MOD-5 calculations is given in Table III for the two basic systems where all calculations were carried to the point where the probability that a neutron had leaked from the system, been absorbed or had dropped below the lowest energy was .999. In all cases, the higher harmonics were still providing a small contribution to the total population at times approaching 300ns, but the major contribution was due mainly to the fundamental.

## C. ANALYSIS OF EXPANSION METHODS

The adequacy of the truncated Fourier series to represent the various source geometry conditions was evaluated by the comparison of the source strength parameter which was determined via the form:

$$\Delta_o = 1.0 - \int_{-x_o}^{x_o} \sum_{1}^{N} (A_n \text{ Cos Kn } x + \text{Cn Sin Kn } x)\, dx$$

$$\Delta_o = 1.0 - \sum_{1}^{N} (-) \frac{An2}{kn} \text{ Sin Kn } x_o - \frac{Cn}{kn} \text{ Cos Kn } x_o .$$

The uncorrected source strength error ($\Delta_o$) was evaluated for the first 100 terms of the series expansion; which stabilized to almost constant-values after the first 30 to 40 terms.

| System | Mode | Time (ns) | Final Neutron Probability Components (Percentage) | | | |
|---|---|---|---|---|---|---|
| | | | Slowing to Bottom of Spectrum** | Leakage | Capture non-fission | Fission |
| Beryllium | 1 | 16620 | 83.9 | 7.4 | 7.9 | - |
| | 2 | 9834 | 44.7 | 48.0 | 6.8 | - |
| | 3 | 4983 | 14.2 | 80.3 | 5.4 | - |
| | 4 | 1952 | 3.1 | 92.6 | 3.4 | - |
| | 5 | 1315 | .01 | 95.6 | 3.4 | - |
| | 6 | 91 | .002 | 96.1 | 2.9 | - |
| ZPR-3 (6F) | 1 | 311 | .95 | 51.1 | 8.7 | 39.2 |
| | 2 | 311 | .001 | 89.2 | 1.2 | 9.5 |
| | 3 | 311 | .0 | 88.6 | 1.2 | 10.2 |

** Bottom of Spectrum is defined as:

Beryllium Slab below 2.31 eV

Assembly 6F below .4 eV

TABLE III. Summary of MOD-5 Computer Runs.

Uncorrected Source Error ($\Delta_o$) 100 Term

| | Source Type | ZPR-3 (6F) | Beryllium Slab |
|---|---|---|---|
| 1- | Square Pulse | 0.0652 | 0.0265 |
| 3- | Exponential | 0.0231 | 0.190 |
| 5- | Ramp Function | 0.0090 | 0.0090 |

The Fourier series expansions are defined such that with an infinite number of terms, the error should go to zero. However, calculations show that for the source function approximations, this error goes to some non-zero value. It should not be any surprise that these source terms have a non-zero error, since the Fourier approximation is based on the interval, $x_o+\delta$, where the distance "$\delta$" represents the extrapolation distance correction for the boundry condition of forcing the neutron density to zero at that point.

In the source function testing, the physical limits of the slab are within a width $2x_o$, rather than boundries of $2(x_o+\delta)$. This additional value of twice the extrapolation distance is responsible for apparent constant error. When the integration is done within the bounds of the slab, a loss occurs at the end points in the intervals

$$-(x_o+\delta) \leq x' \leq -x_o \quad \text{and} \quad +x_o \geq x \geq (x_o+\delta).$$

As the ratio $\delta/x_o$ is greater than zero for all real (non-infinite systems), this apparent difficulty in the approximation will be almost negligible for systems with the slab width much greater than the extrapolation distance ($\delta/x_o$<<1.0).

Each of the three main source types (the square pulse, exponential and ramp functions) were studied to the limit of

a 100 mode harmonic expansion for the spatial dependent functions. The value of the delta correction ($\Delta_o$) are shown in Figures 5 and 6 for both systems. While all test calculations were done for a six mode expansion, the effect of the truncation to less than an infinite series approaches an asymtotic limit in each source type, the limits on the value of $\Delta_o$ are a measure of the $\delta/x_o$ ratio. In both of the systems studied, it can easily be seen that inclusion of more than 10 harmonic modes would not significantly change the spatial dependent values.

## D. COMPARISON OF SOURCE TYPES

The development and testing of the time-space-energy dependence for five source geometries was a task more formidable than originally anticipated. Numerous programming difficulties arose toward the end of the project for the two cases of the off-axis sources, which resulted in their not being included in the final calculations. Both of these sources require modification of the spatial harmonic functions than those proposed previously.

The three source conditions that are included in the detailed calculations were the symmetrical square pulse and the first collision source centered about the mid-point of the slab and the exterior wide beam or ramp function source.

The existing version of the computer program, MIL-6, demonstrated the effectiveness of the three main source conditions, but the off-axis square pulse and first collision geometry must be revised and retested prior to further analysis.

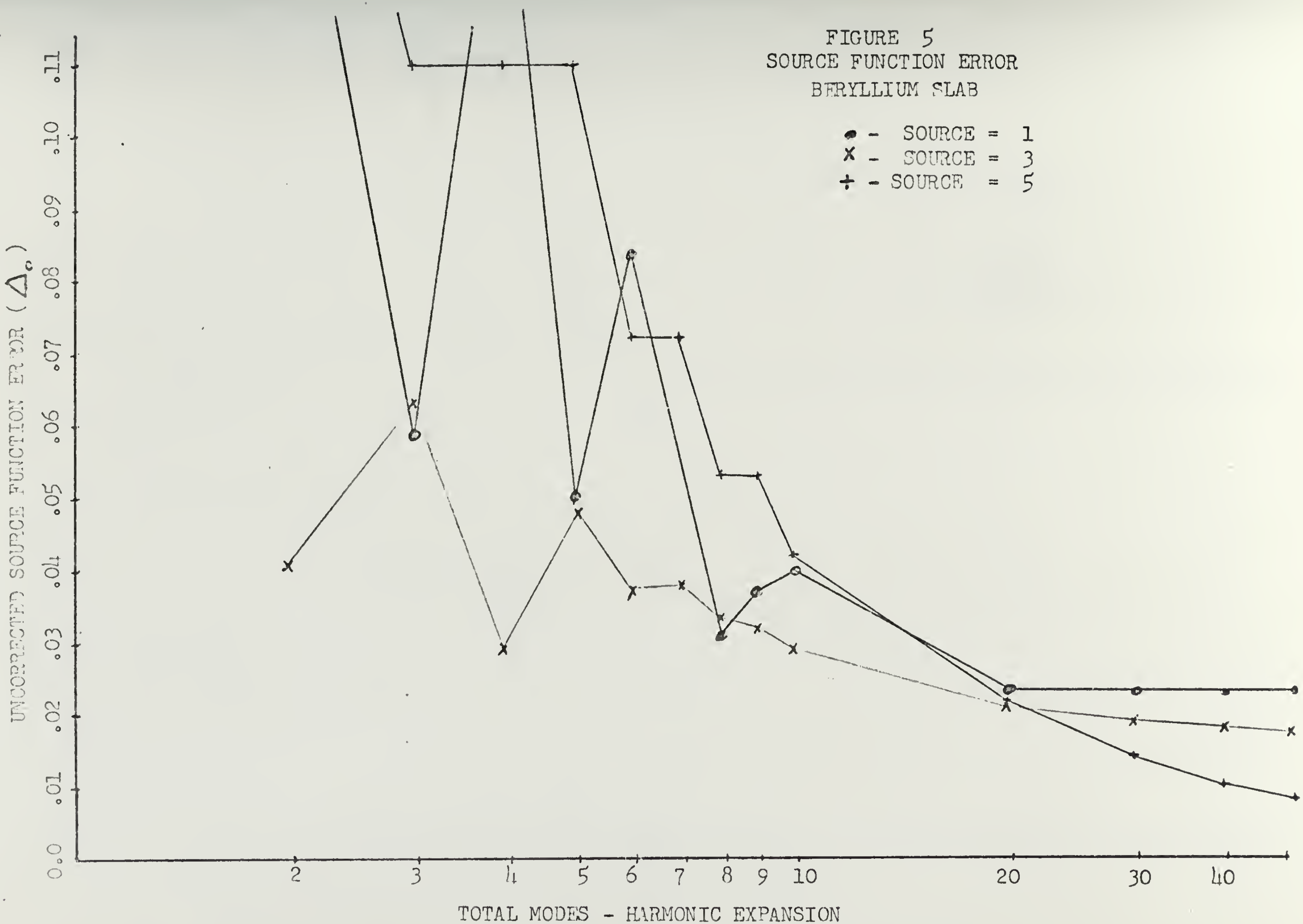


FIGURE 5
SOURCE FUNCTION ERROR
BERYLLIUM SLAB
- SOURCE = 1
X - SOURCE = 3
+ - SOURCE = 5
UNCORRECTED SOURCE FUNCTION ERROR ( $\Delta_o$ )
0.0
.01
.02
.03
.04
.05
.06
.07
.08
.09
.10
.11
2
3
4
5
6
7
8
9
10
20
30
40
TOTAL MODES - HARMONIC EXPANSION

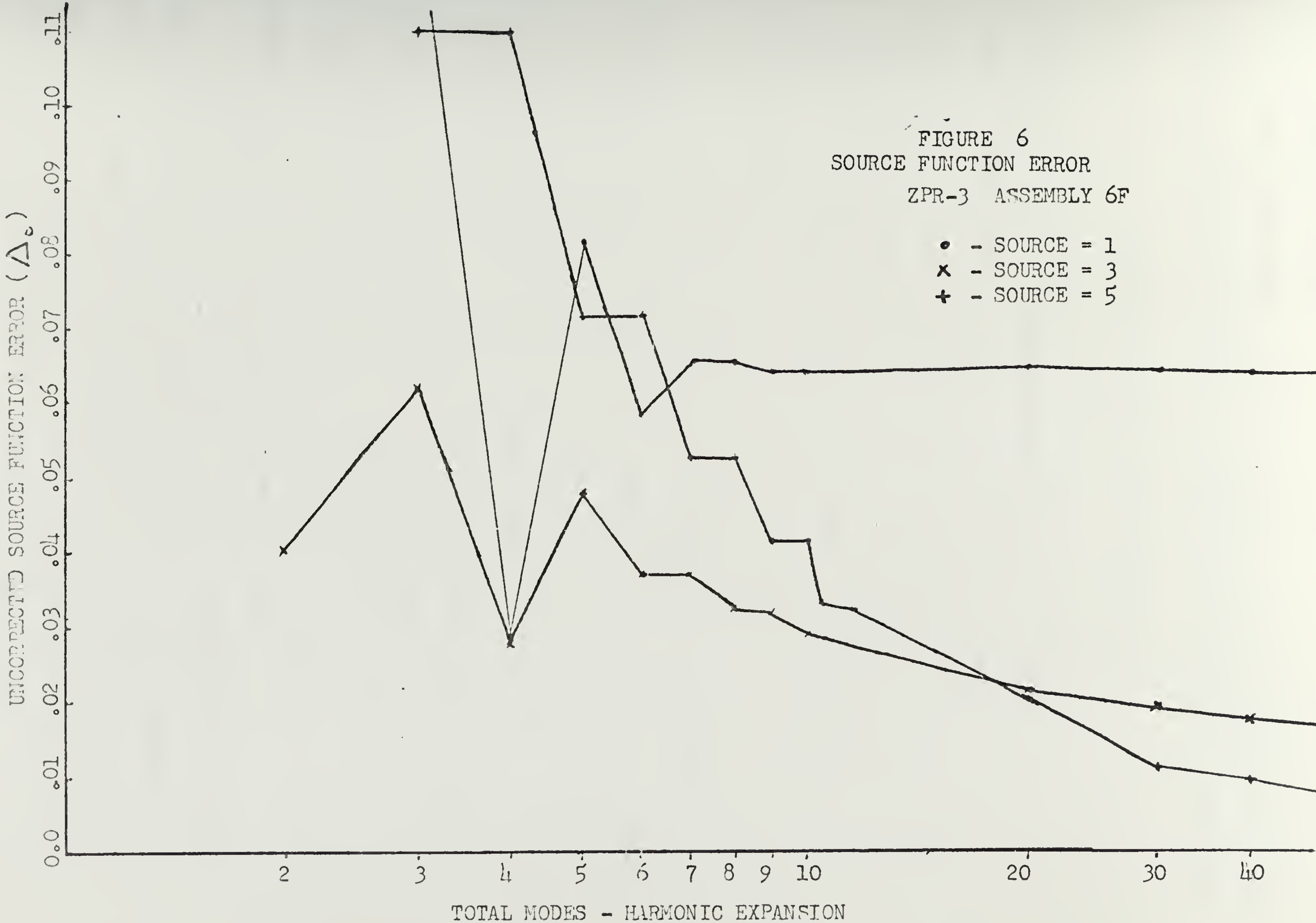


FIGURE 6
SOURCE FUNCTION ERROR
ZPR-3 ASSEMBLY 6F
• - SOURCE = 1
× - SOURCE = 3
+ - SOURCE = 5
UNCORRECTED SOURCE FUNCTION ERROR (Δ꜀)
0.0
.01
.02
.03
.04
.05
.06
.07
.08
.09
.10
.11
2
3
4
5
6
7
8
9
10
20
30
40
TOTAL MODES - HARMONIC EXPANSION

## E. NEUTRON SPECTRA

The primary goal of this project was to demonstrate the spatial dependent response of the neutron spectral function:

$$N(x,v,t) = \sum_n R_n(x)\ F_n(v,t).$$

The contribution of each harmonic mode to the neutron spectra is determined by a Fourier space harmonic term, $R_n(x)$, and the MOD-5 time-lethargy probability density for the n-th time mode, $F_n(v,t)$. For each harmonic mode, the probability density function is subject to the basic condition:

$$F_n(v,t) = 1.0 \ - \text{(Absorption + Leakage)}$$

and the initial neutron source spatial distribution provides the correct source contribution.

## F. NEUTRON DENSITY AND NEUTRON FLUX

The results of neutron density and neutron flux calculations are presented at a limited number of points to be representative of the over all processes, however, inclusion of all points in the slab would result in an inordinate number of graphs and tables of data.

The neutron density, $N(x_k,v_i,t_j)$, was plotted only for the point $x_k$=0.0 and a complete spectral plot is given in Figures 13 and 16, to be representative of the information possible to determine. All other calculations made by the program MIL-6 required the evaluation of the density at all points in the slab, and similar response curves were prepared.

The total neutron flux,

$$\phi(x_k,t_j) = \sum_i v_i \, N(x_k,v_i,t_j),$$

is presented for one location half way between the center and the surface of the slab, the same positions selected for the detector response calculations. Each plot is for a single system, either the Beryllium or assembly 6F, follows the complete time decay of the total flux from one nanosecond to the time at which the neutron population had decayed to 0.1% of its initial value. In both cases, the first build up of the flux indicates the initial stages of the neutron motion within the slab, to the point at 20ns at which the leakage from the free surface becomes the dominant effect.

When the leakage terms are considered, the appearance of the decay curve now looks almost identical with the pseudo decay modes previously described. When looking at the neutron density curves, it can easily be noticed that it required several nanoseconds for the initial 2.46 MeV burst to develop into a smooth distribution.

## G. MEAN ENERGY AND MEAN ENERGY RATIO

A convenient method of comparing the time and spatial effects on the neutron density is by comparing the average or mean energy of the neutrons at all points in the slab at the same time. A complete history of the neutron population can be concisely presented by observing the absolute variation of the mean energy at a single reference point in

the slab and then comparing the relative changes in the entire slab in terms of the reference point for each discrete time step.

The basic reference point for all comparisons is the center of the slab, x=0.0, for all geometries and systems. The decay of the mean energy with time is shown for both beryllium (Figure 19) and assembly 6F (Figure 20). Starting with the two reference cases, the spatial variation from the center to the surface of the slab starting at two nanoseconds after the pulse to the final decay of the fundamental mode, (beryllium - Figures 21 to 30--assembly 6F Figures 31 to 36).

After the initial pulse, a wave-like shape is observed to develop across the slab in the direction of the free surface, as one would expect the higher energy neutrons to migrate out of the system fastest. This is shown most vividly in Figures 24 to 27.

Near the end of the decay period, the last remaining neutrons in the system have almost a constant ratio of one, as the system tends toward an equilibrium condition. The leakage at the surface allows the final stages to show this condition (Figures 30 and 36).

## H. DETECTOR RESPONSE

This computational routine did not work as well as intended. Minor computational and programming errors were encountered very late in the project and have not been completely resolved. These will require revision and further testing as follow on work.

Comparisons of the flux calculations and the mean energy determinations referred to in Sections F and G were compared to the "proton recoil" detector (Figures 37 and 38) which follow the same basic trends for the time dependent response, but the numerical differences in the time integrated response will require improvement of the space integral and time summation numerical methods.

# VI. RECOMMENDATIONS

## A. FOLLOW ON WORK

The program MIL-6 was prepared to be a general computational tool using the harmonic mode expansion method for the one dimensional systems. Principal follow on work is recommended to correct the minor computational difficulties encountered, write and test data transfer routines for both MOD-5 and MIL-6 and review and edit the programming for a more optimized execution.

Principal areas to have future work are:

(1) re-evaluate the Fourier space functions for the two off-axis source geometries,

(2) revise and test the numerical integration techniques of the detector response program for the summation over time, and consider methods to improve the space integration over the detector volume,

(3) consider a detector response function that would be more realistic than the STP hydrogen case now included in the program,

(4) complete and test the input/output options using nine-track magnetic tape and the IBM-2314 magnetic disc systems for data transfer between the programs MOD-5 and MIL-6,

(5) revise the program MIL-6 to handle more than six harmonic modes by use of a smaller state structure of 70

to 100 states rather than the 150 states the program is currently written to process,

(6) provide more realistic source geometries for the one dimensional case than the five simple ones currently in the program,

(7) review the program for reduction in the core storage space requirements and to improve the speed of execution through program optimization.

## B. IMPROVEMENTS TO PROGRAM MIL-6

The original program MIL-6 was written to deal with monoenergetic neutron sources and delta functions in time. This may require significant modification of the computational methods to considering the fission spectrum sources and sources that have a finite width time duration. These situations had not been tested in the MIL-6 work and would provide a more realistic model for fast neutron experiments.

A second item that appears worthy of consideration in the future will assist in modeling the wave-like motion of the neutron pulse from the finite volume square pulse and the exterior wide beam ramp function. This could be considered as a limiting correction factor to couple the time-space and energy dependence during the initial response to a neutron pulse.

## C. THREE DIMENSIONAL SYSTEMS

The harmonic expansion method has been demonstrated to work for the one dimensional case, and definite action to

begin study of three dimensional systems is warranted. A recommended sequence of priorities would be to expand the MIL-6 work to three dimensions in the cartesian coordinates first, cylindrical geometry second and spherical geometry coordinate systems last.

## APPENDIX A: NUMERICAL RESULTS

A. SPECTRAL RESPONSE

1. Beryllium Slab

a. Neutron density harmonic component $N_n(x,v_i,t_j)$ for harmonic modes 1 to 6 at point x=0.0. (Figures 7 to 12.)

b. Neutron density: $N(x,v_i,t_j)$: lethargy - time neutron population density versus lethargy for a 6 mode expansion at point x=0.0. (Figure 13.)

2. ZPR-3 Assembly 6F

a. Neutron density harmonic component $N_n(x,v_i,t_j)$ for harmonic modes - 1 and 2 at point x=0.0. (Figures 14 and 15.)

b. Neutron density $N(x,v_i,t_j)$ - lethargy - time neutron population density versus lethargy - for a sum of 3 harmonic modes at x=0.0. (Figure 16.)

B. MEAN ENERGY AND MEAN ENERGY RATIO

1. Beryllium

a. Mean energy verus time at x=0.0. (Figure 19.)

b. Plot of mean energy ratio versus position for 8 time periods. (Figures 21 to 30)

2. ZPR-3 (6F)

a. Mean energy versus time at x=0.0. (Figure 20.)

b. Plot of mean energy ratio versus position for 6 time periods. (Figures 31 to 36.)

## C. DETECTOR RESPONSE

1. Beryllium - Figure 37.

2. ZPR-3 - Figure 38.

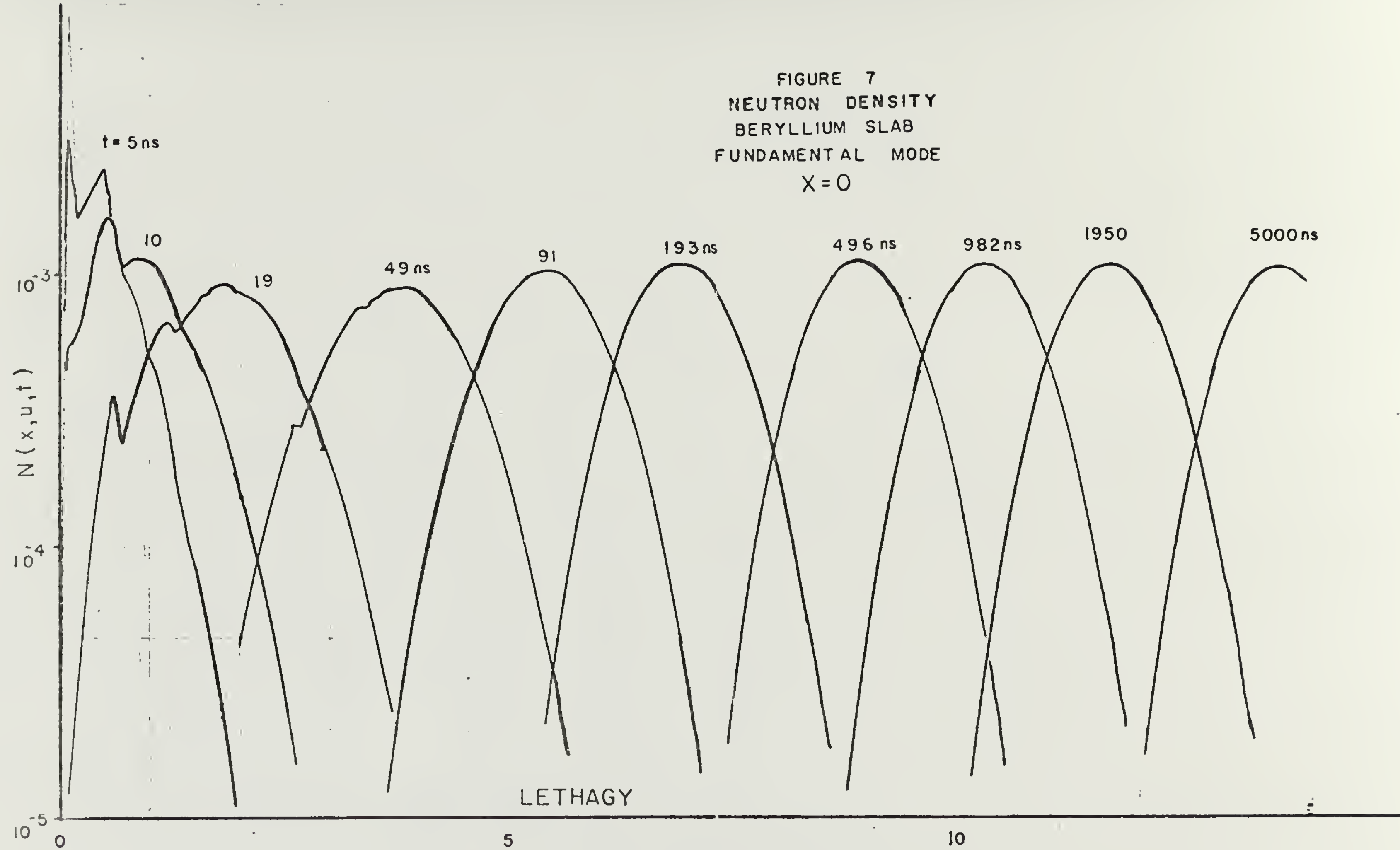



FIGURE 7
NEUTRON DENSITY
BERYLLIUM SLAB
FUNDAMENTAL MODE
X = 0

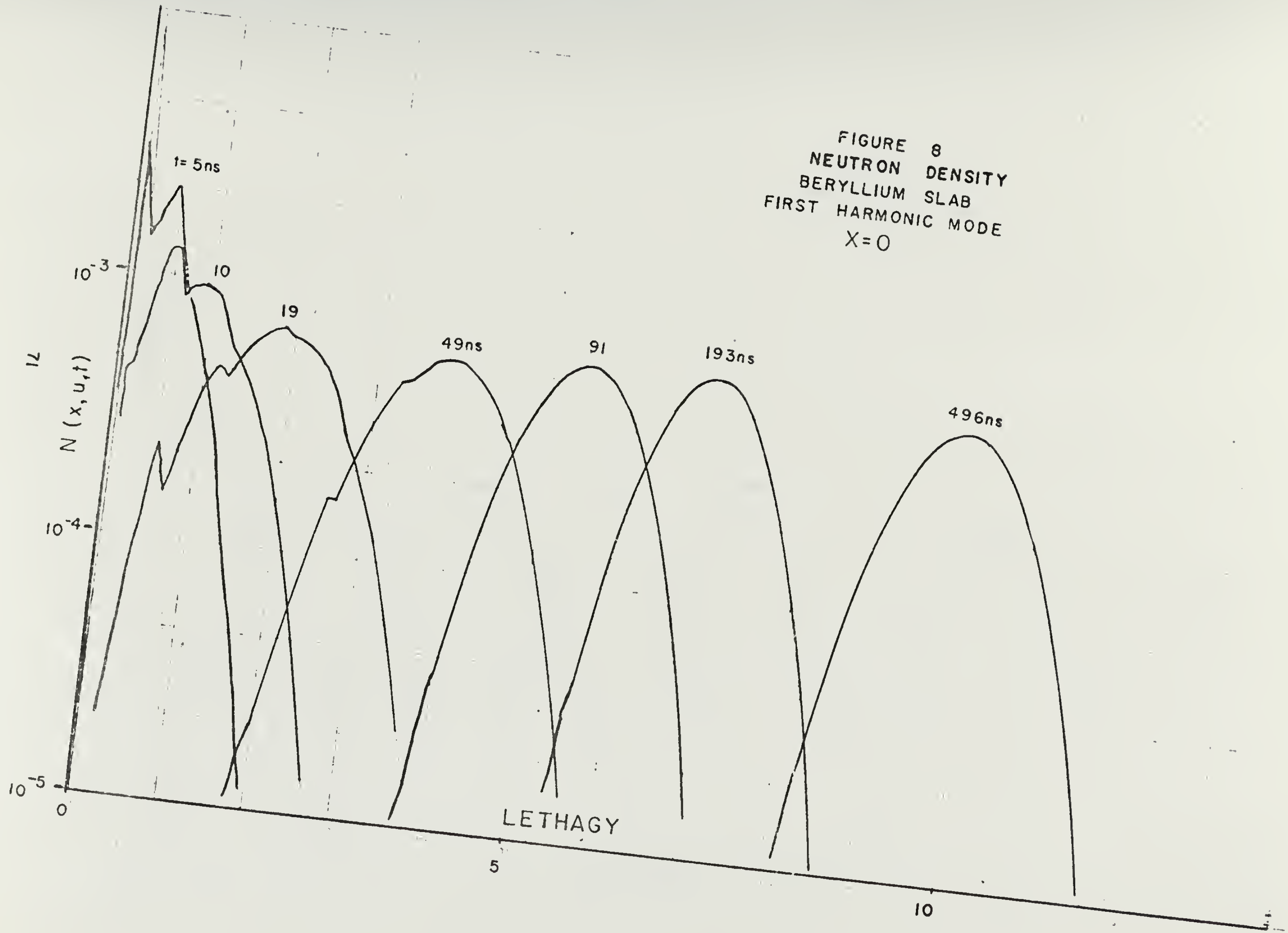



FIGURE 8
NEUTRON DENSITY
BERYLLIUM SLAB
FIRST HARMONIC MODE
X=0

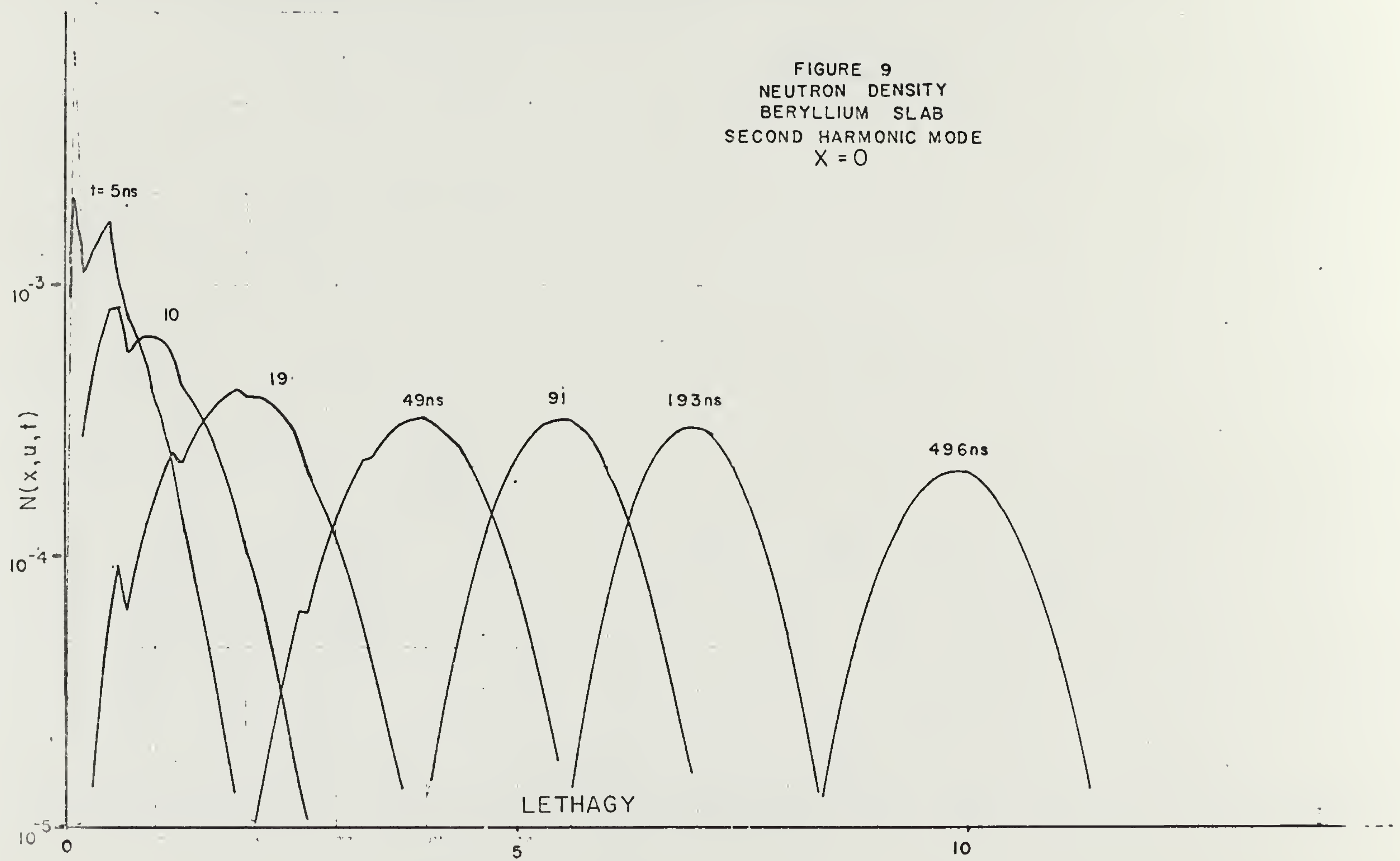
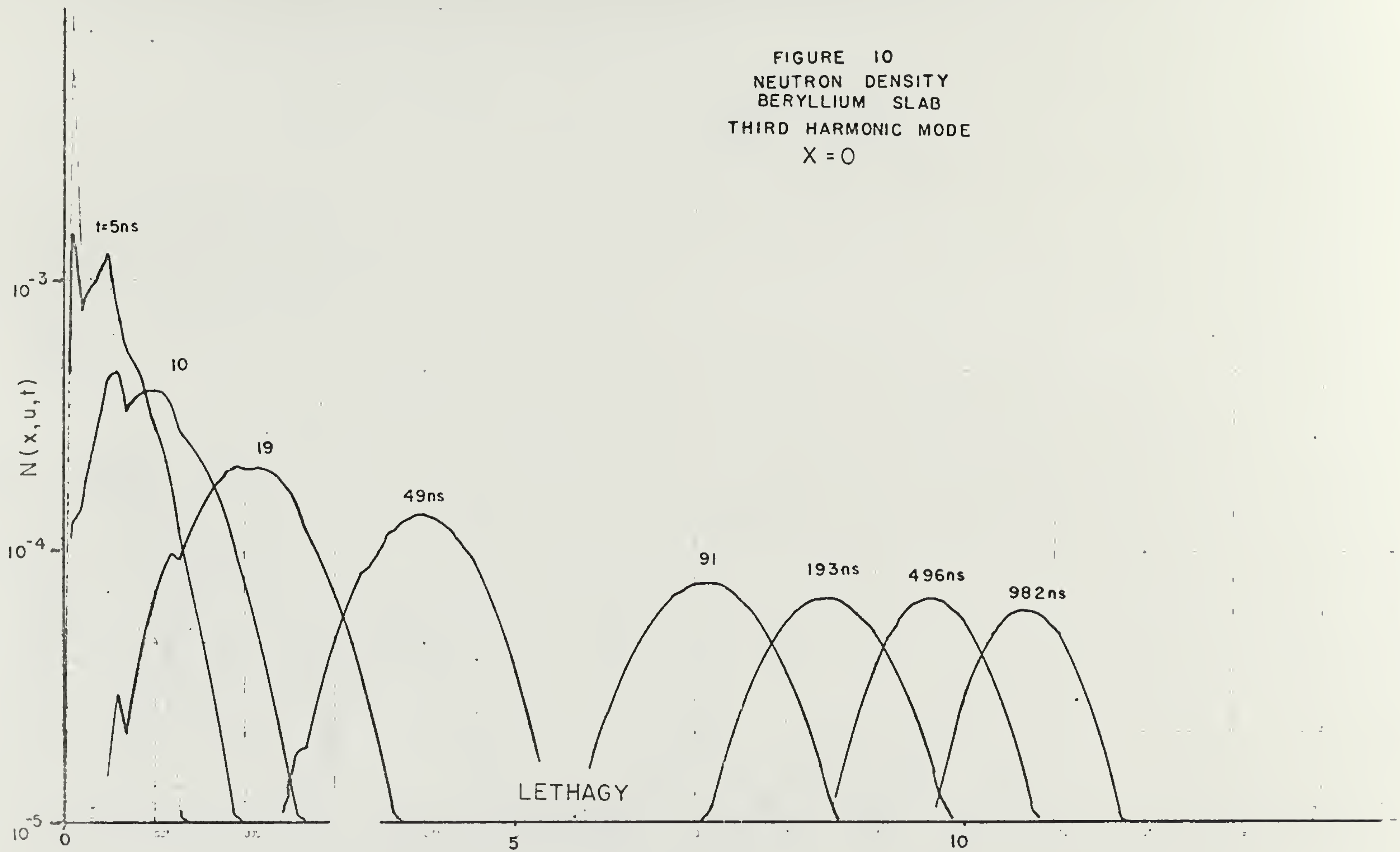



FIGURE 9
NEUTRON DENSITY
BERYLLIUM SLAB
SECOND HARMONIC MODE
X = 0

FIGURE 10
NEUTRON DENSITY
BERYLLIUM SLAB
THIRD HARMONIC MODE
X = 0



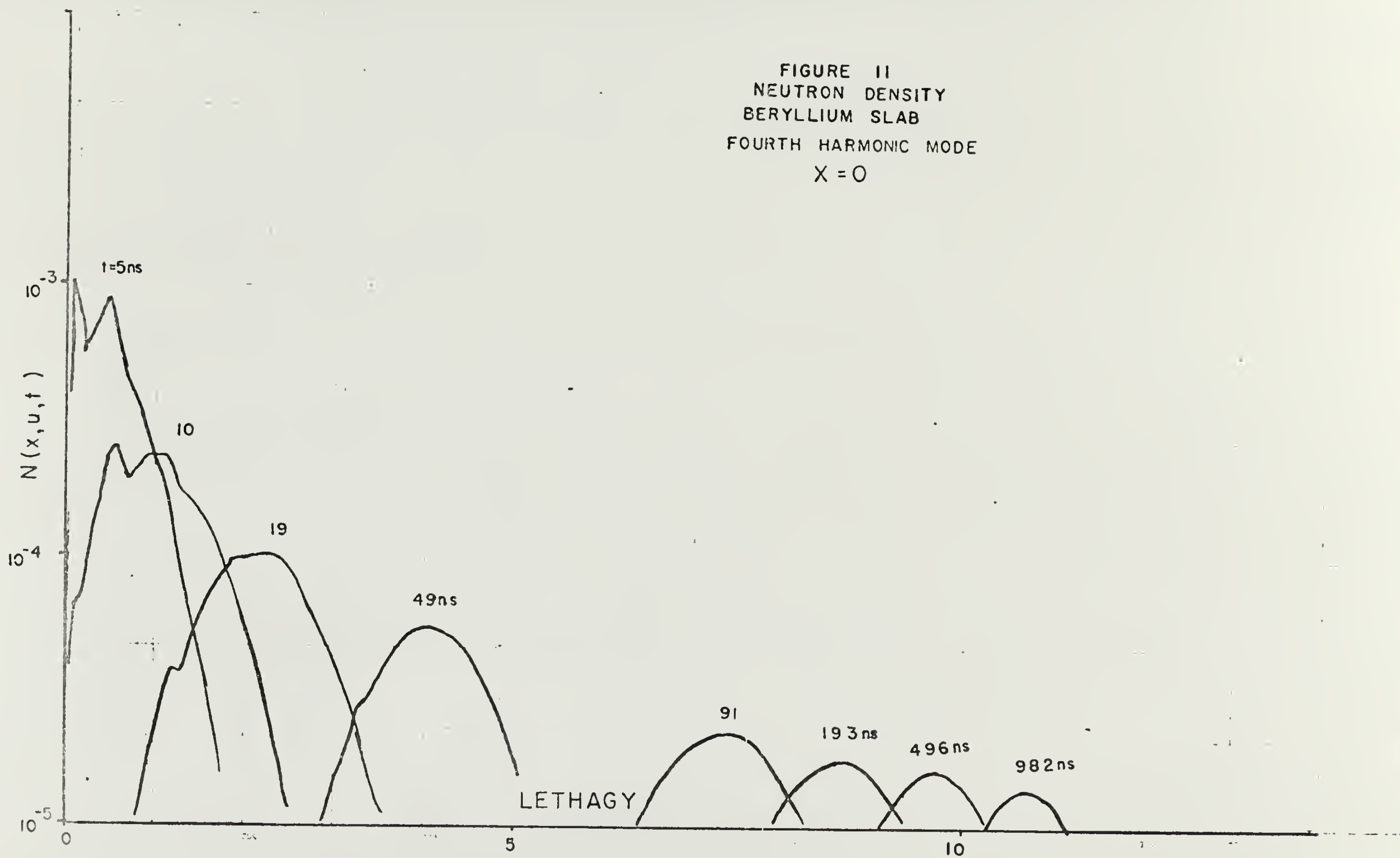
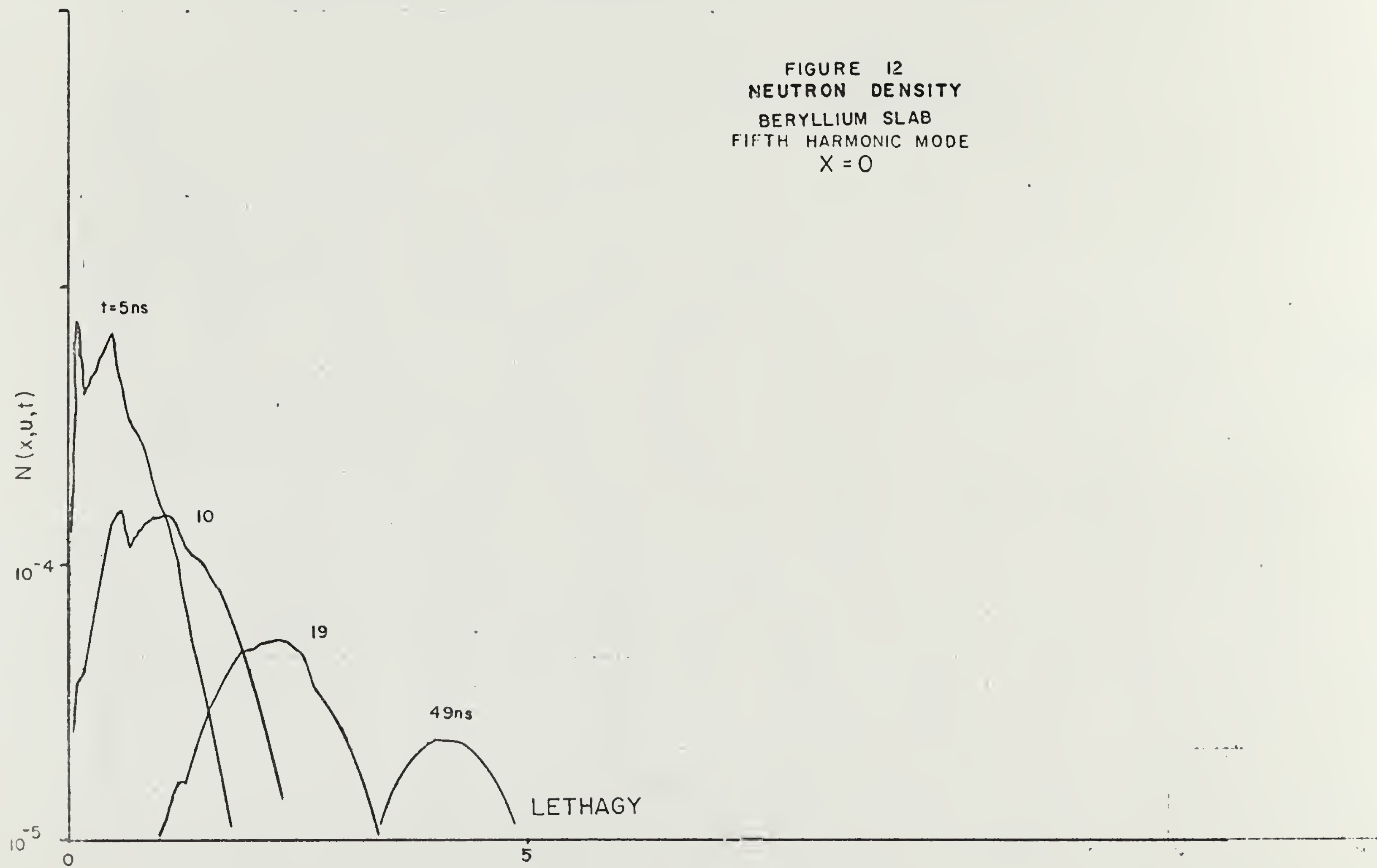



FIGURE 11
NEUTRON DENSITY
BERYLLIUM SLAB
FOURTH HARMONIC MODE
X = 0

FIGURE 12
NEUTRON DENSITY
BERYLLIUM SLAB
FIFTH HARMONIC MODE
X = 0



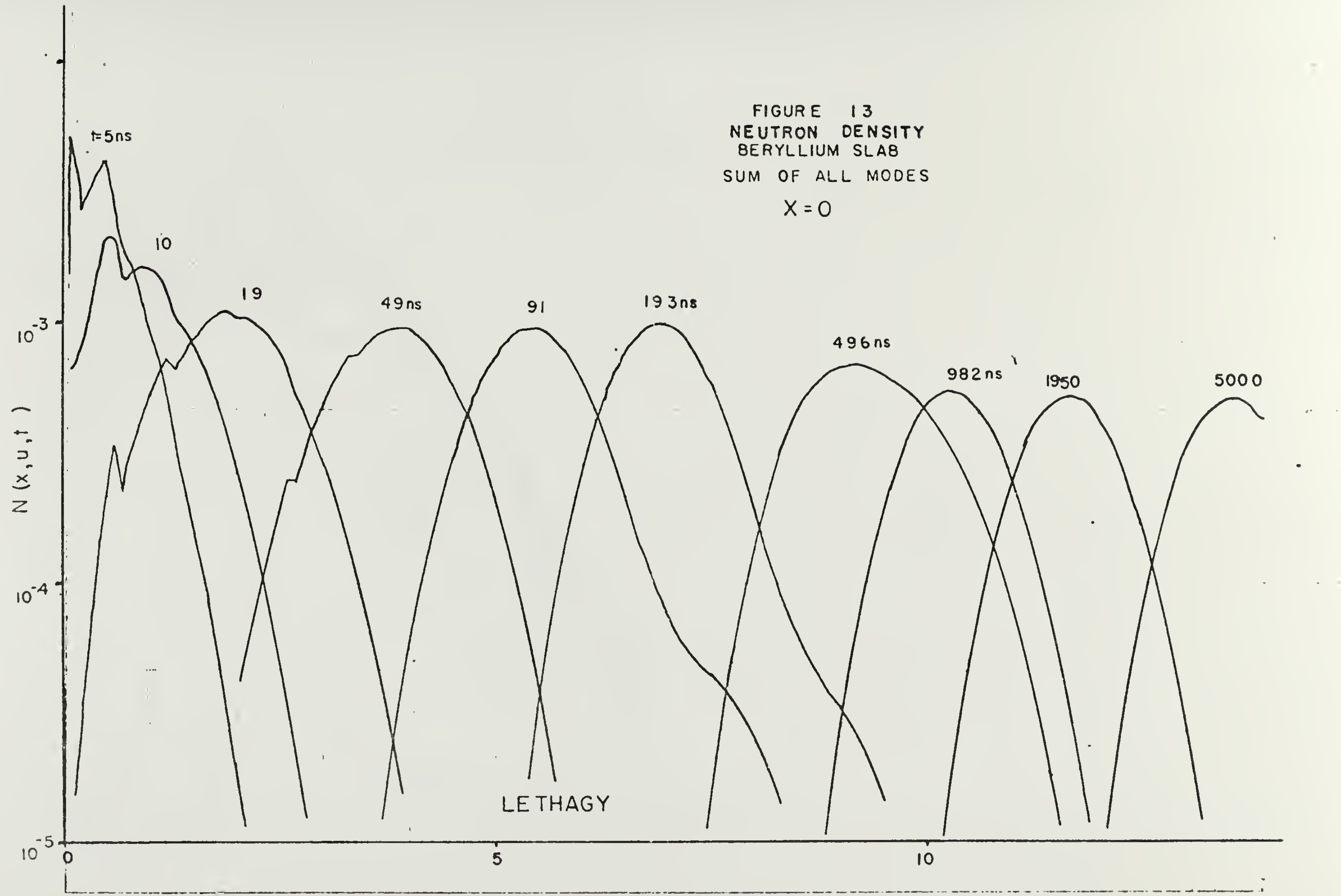



FIGURE 13
NEUTRON DENSITY
BERYLLIUM SLAB
SUM OF ALL MODES
X = 0

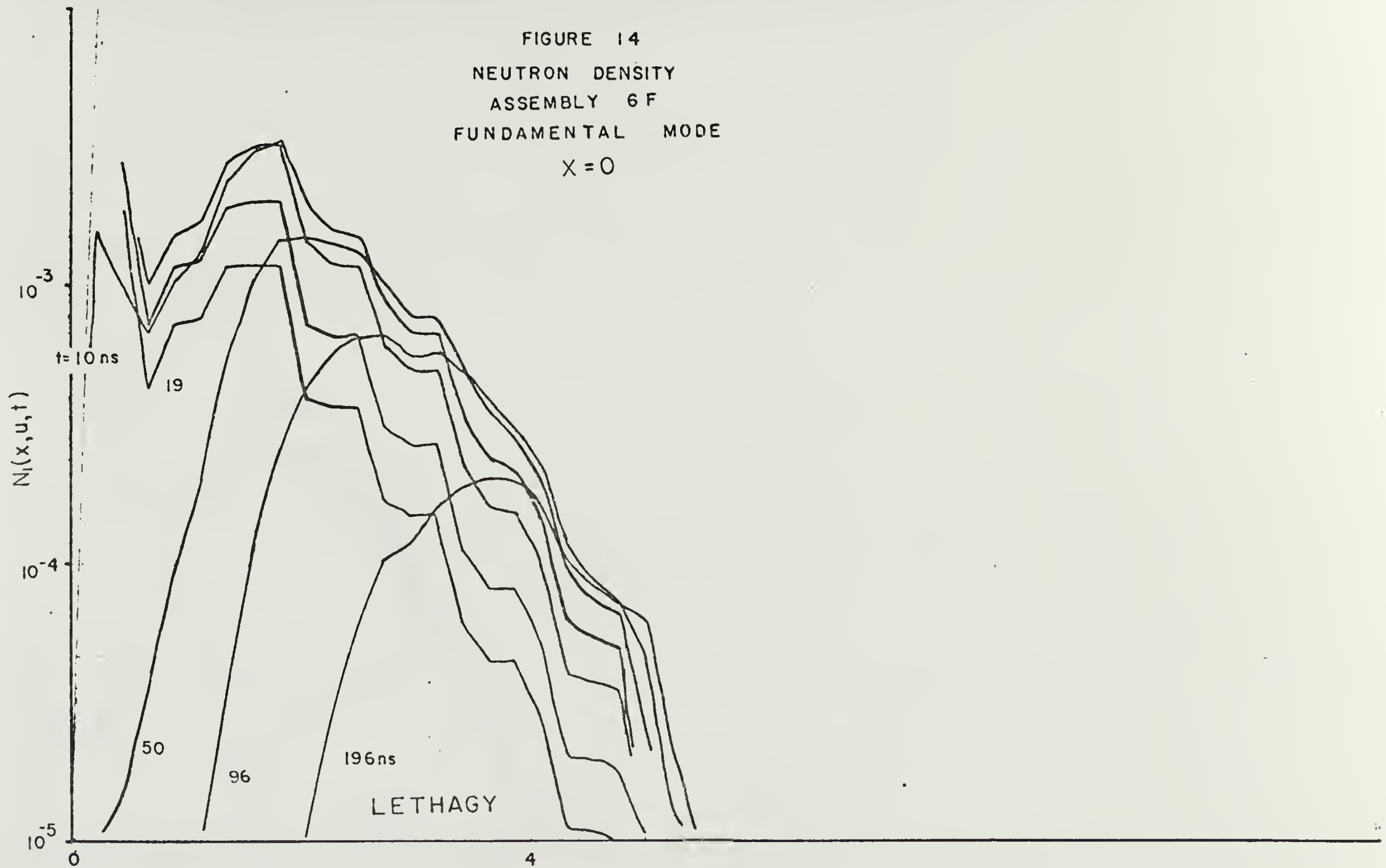



FIGURE 14
NEUTRON DENSITY
ASSEMBLY 6 F
FUNDAMENTAL MODE
X = 0

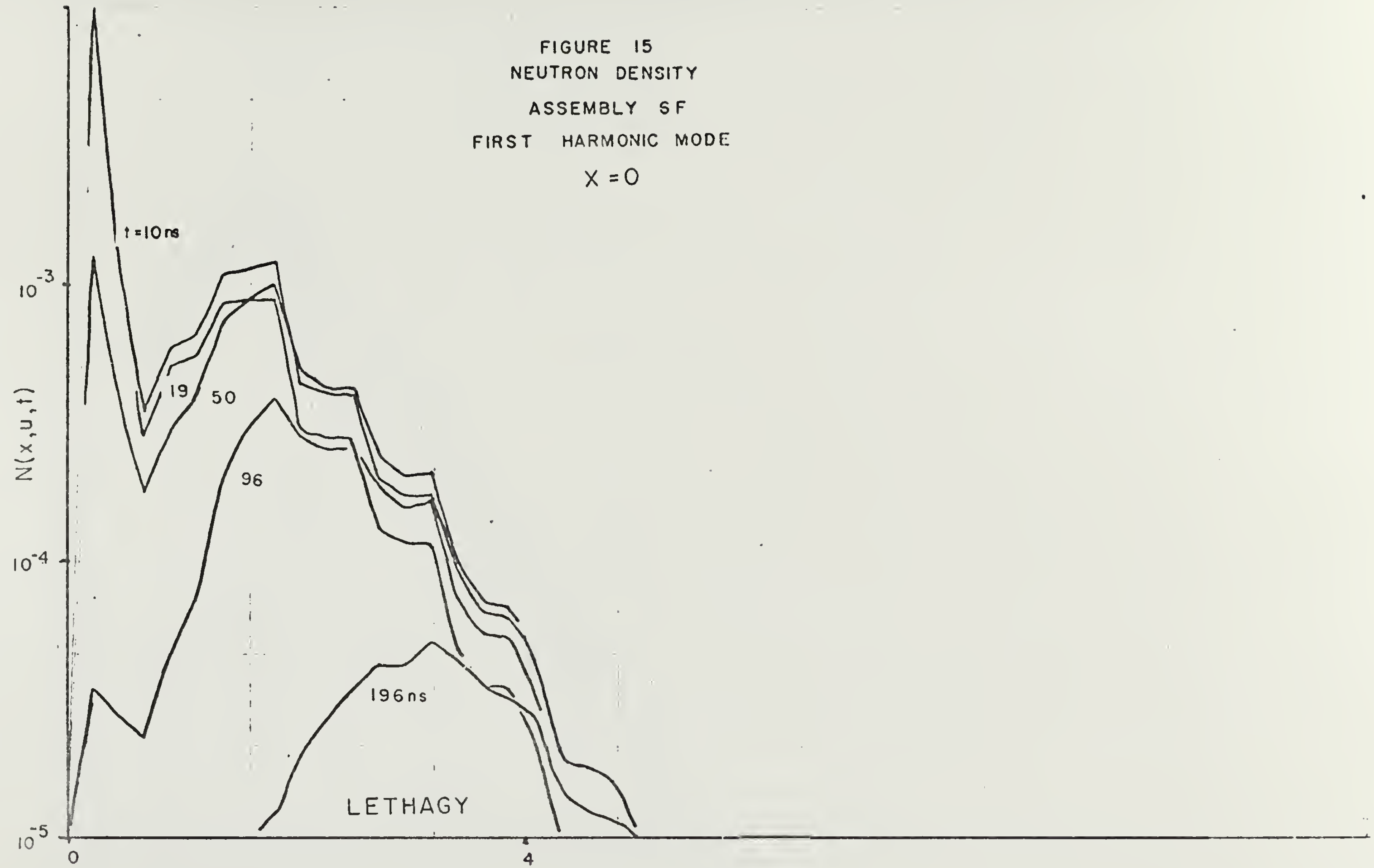



FIGURE 15
NEUTRON DENSITY
ASSEMBLY SF
FIRST HARMONIC MODE
X = 0

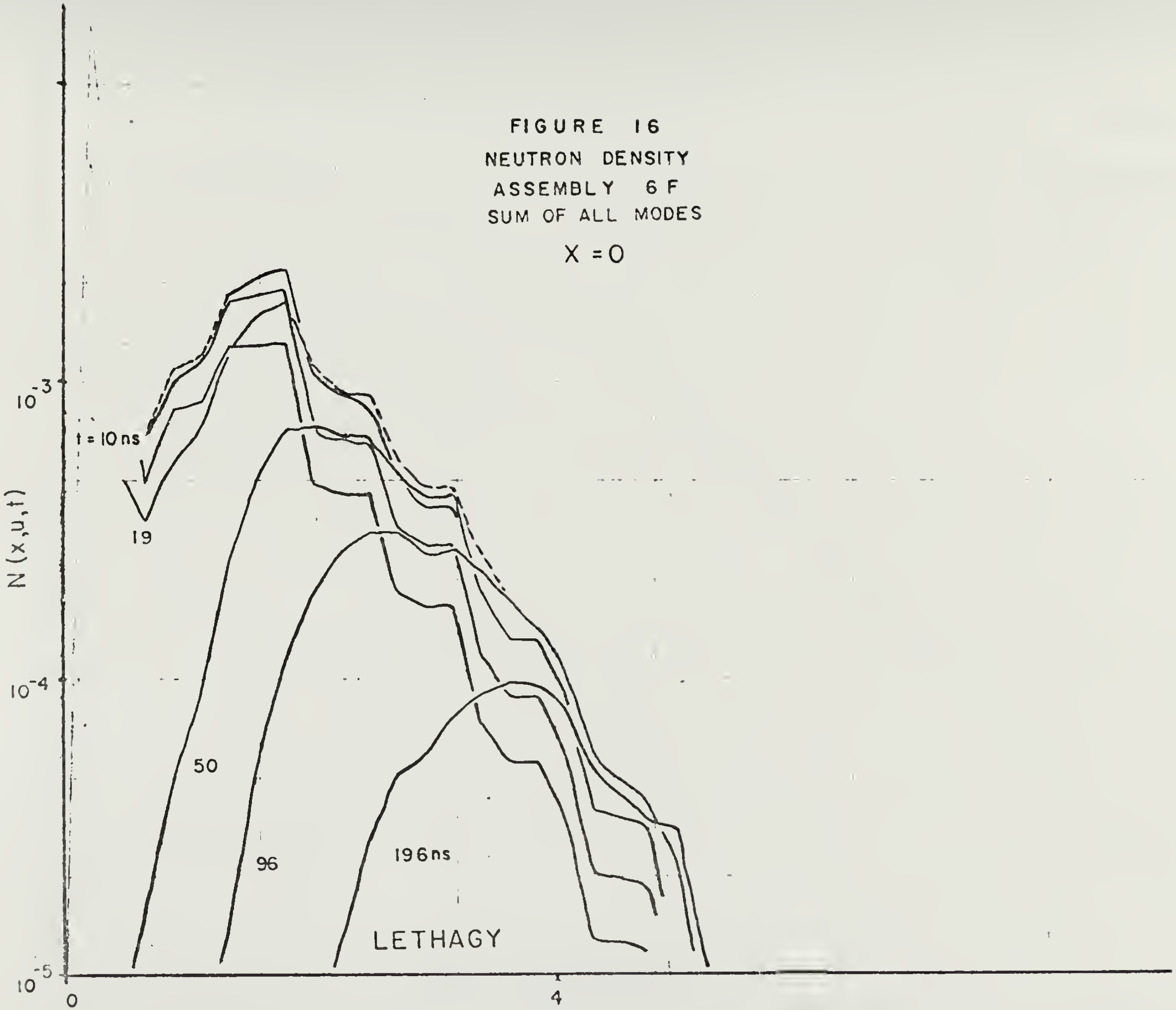
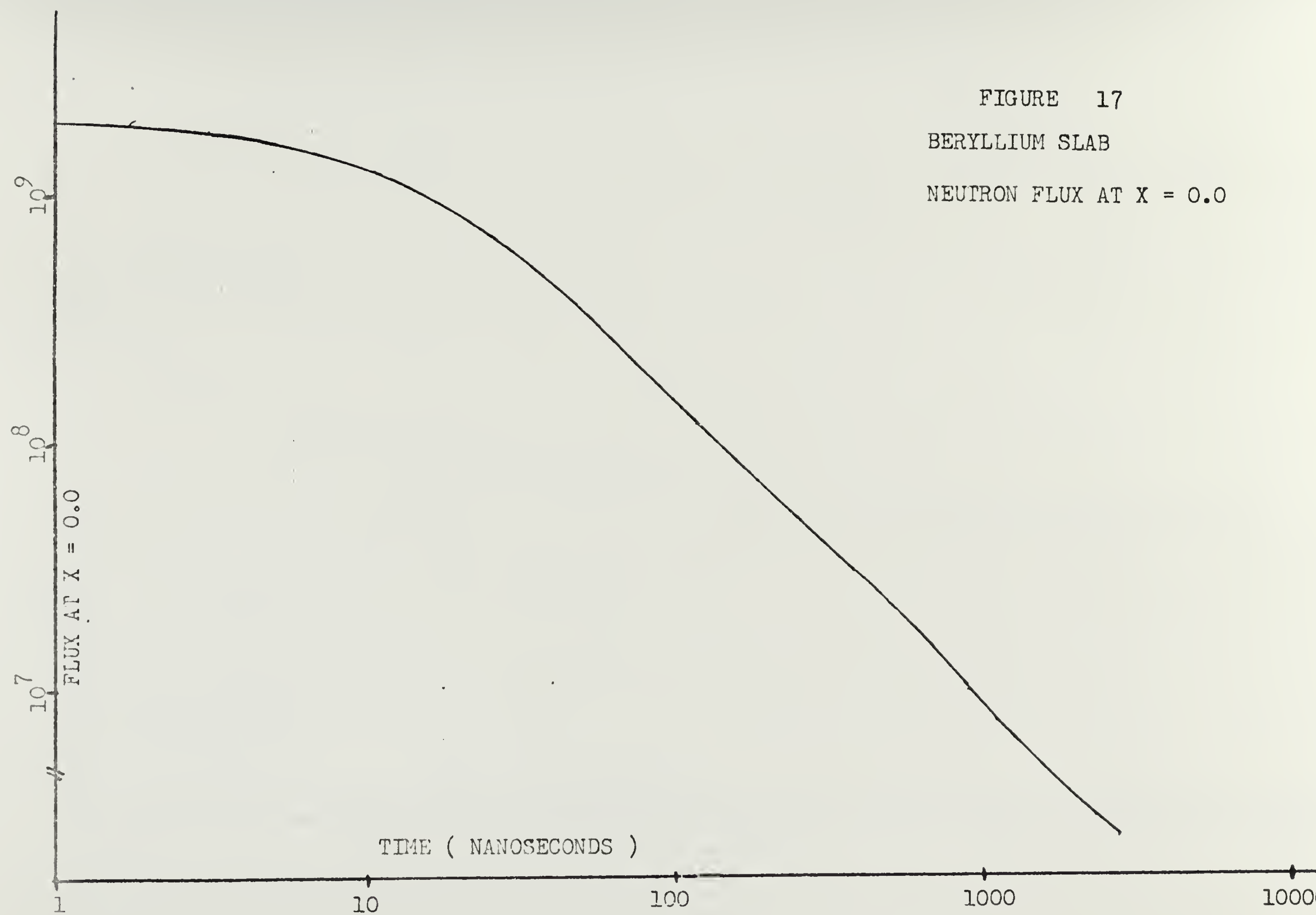



FIGURE 16
NEUTRON DENSITY
ASSEMBLY 6 F
SUM OF ALL MODES
X = 0

FIGURE 17

BERYLLIUM SLAB

NEUTRON FLUX AT X = 0.0



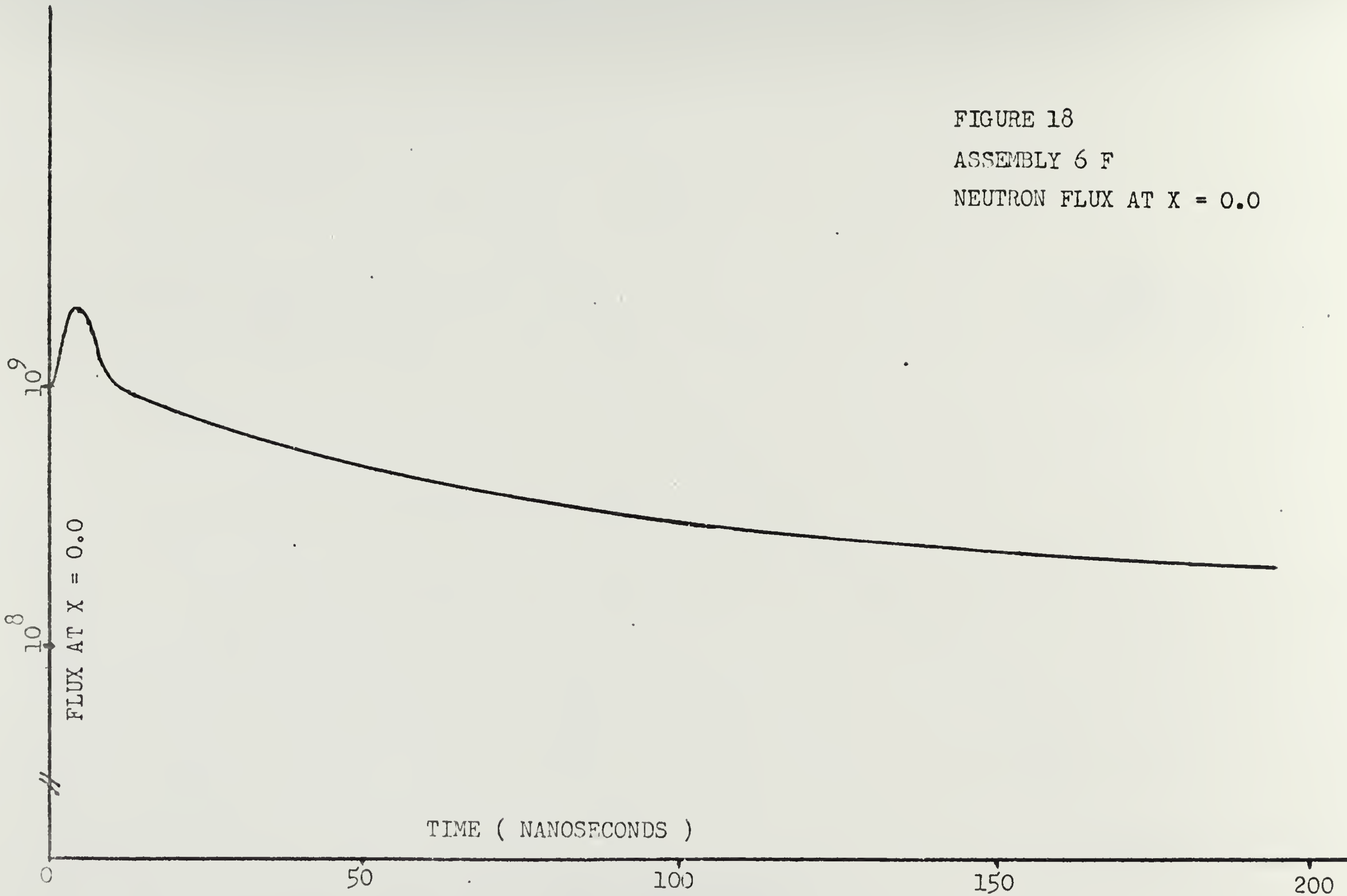



FIGURE 18
ASSEMBLY 6 F
NEUTRON FLUX AT X = 0.0

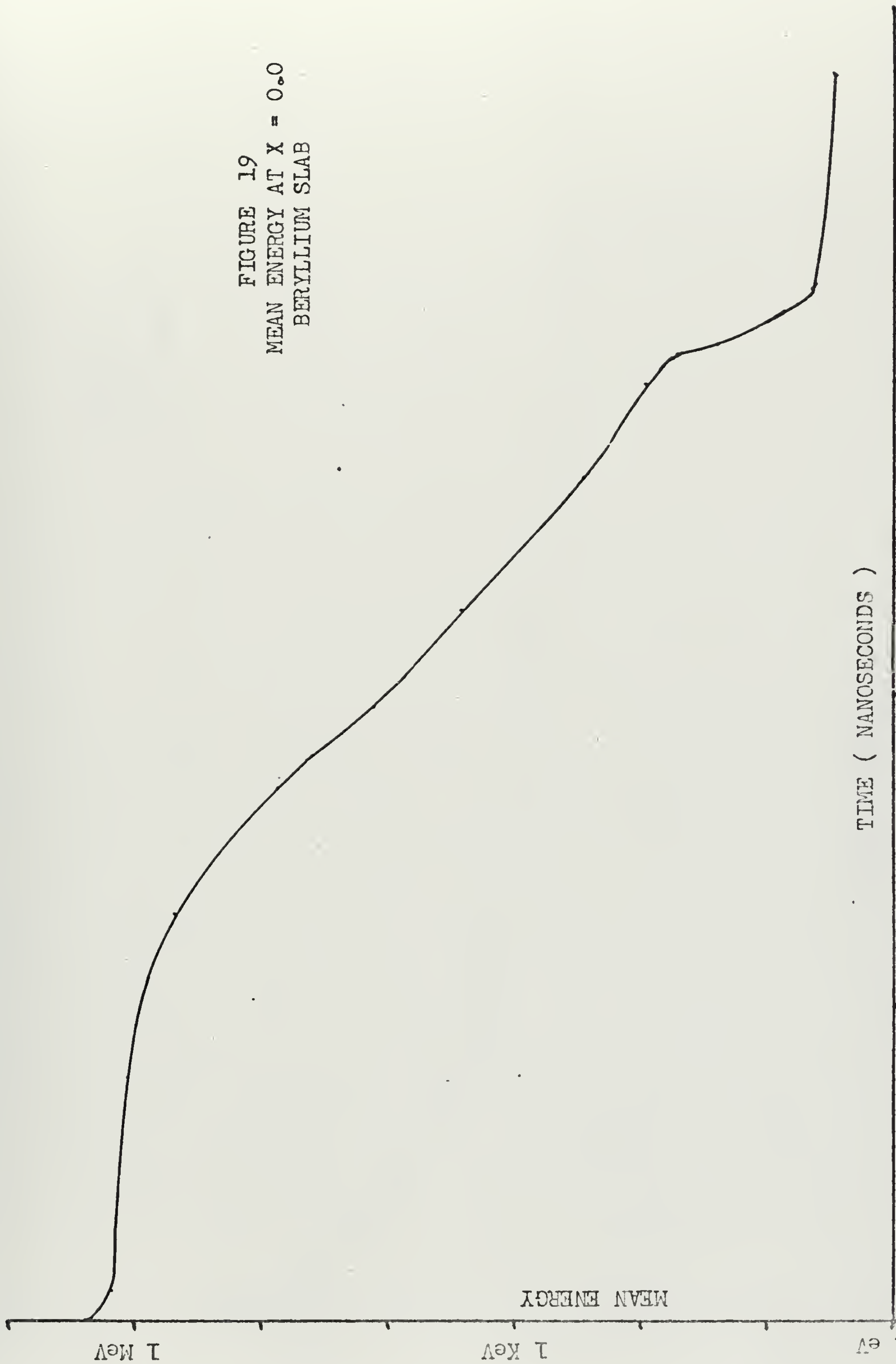
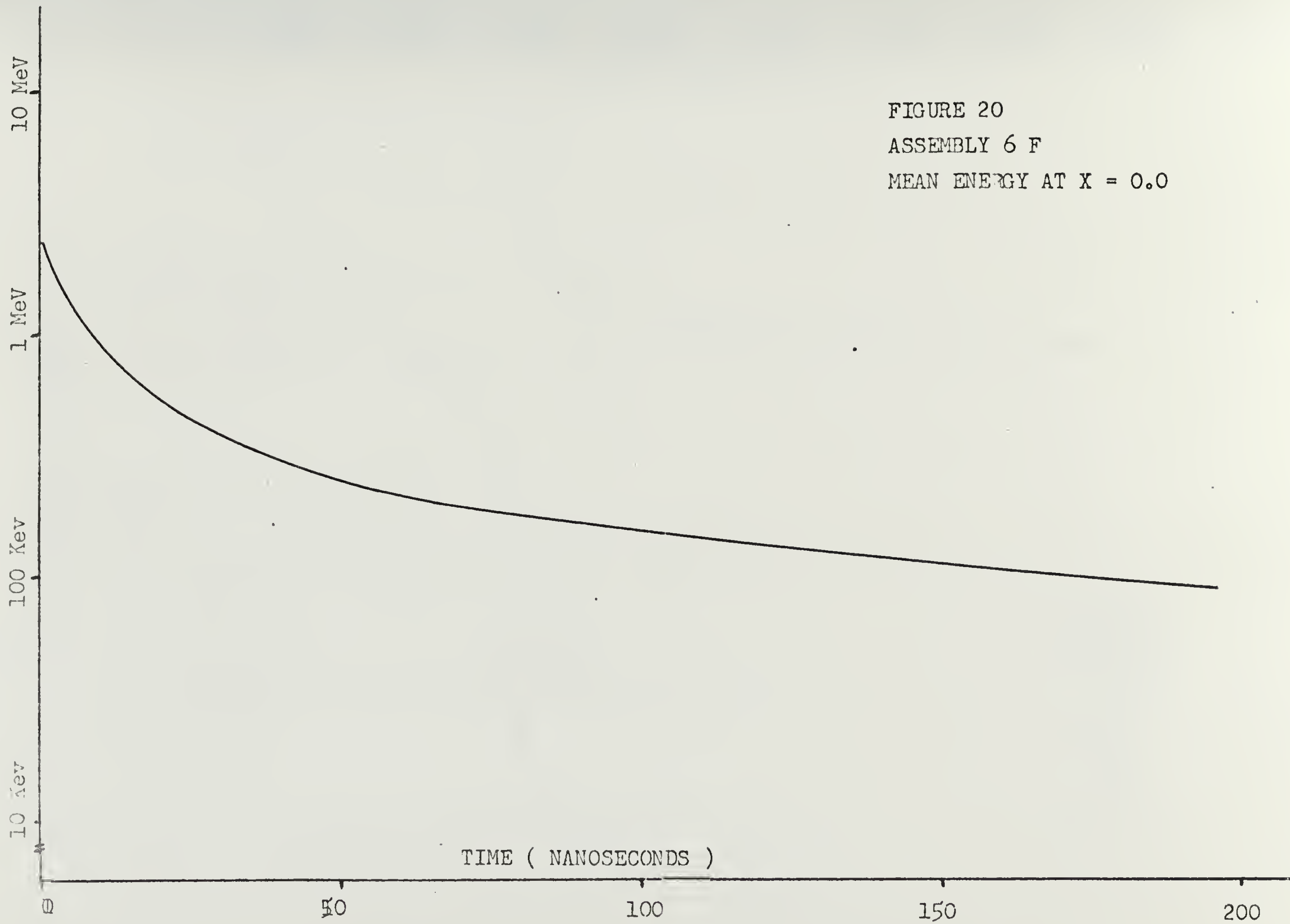



FIGURE 19
MEAN ENERGY AT X = 0.0
BERYLLIUM SLAB

FIGURE 20
ASSEMBLY 6 F
MEAN ENERGY AT X = 0.0



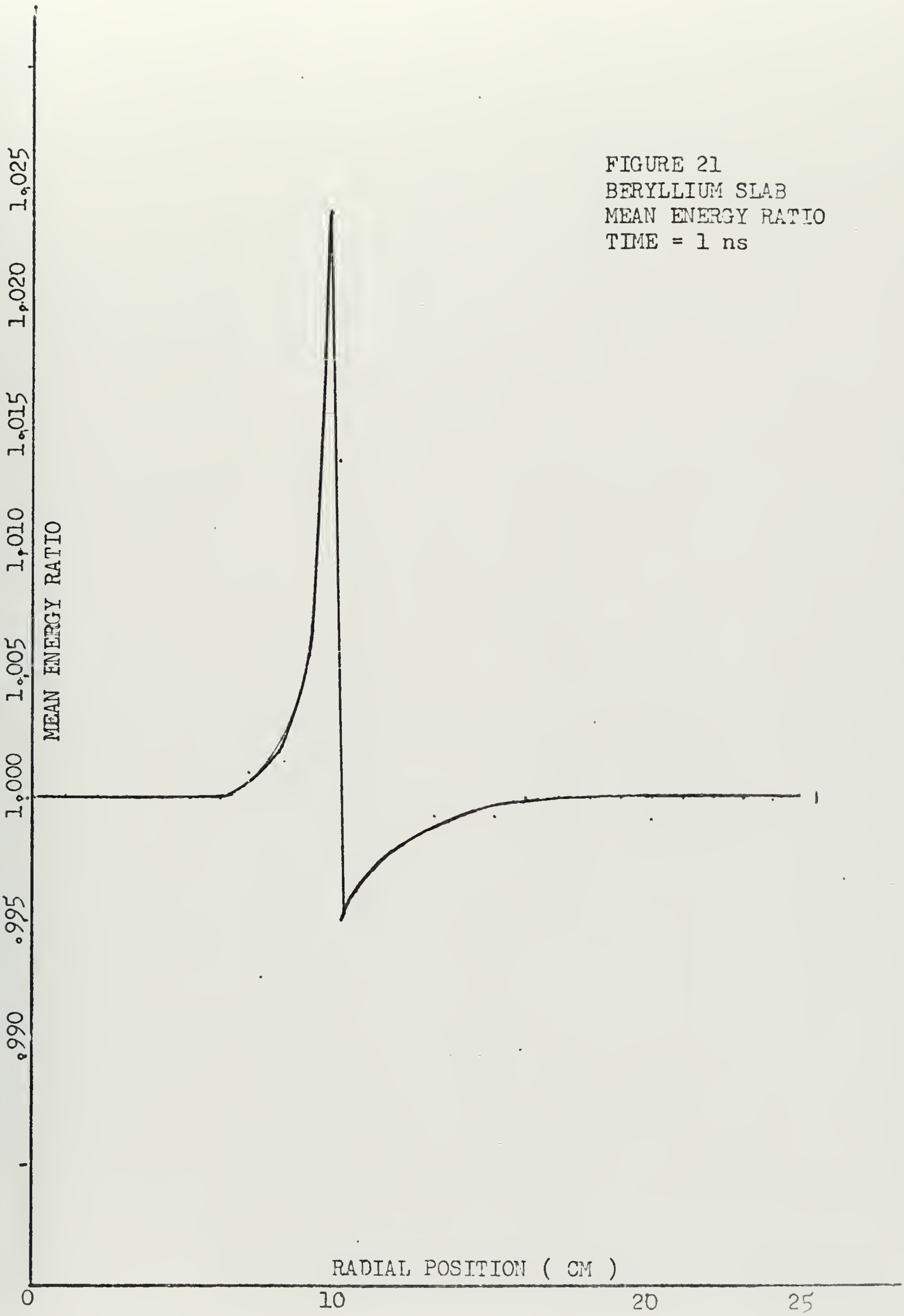


FIGURE 21
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 1 ns
MEAN ENERGY RATIO
1.025
1.020
1.015
1.010
1.005
1.000
.995
.990
RADIAL POSITION ( CM )
0
10
20
25

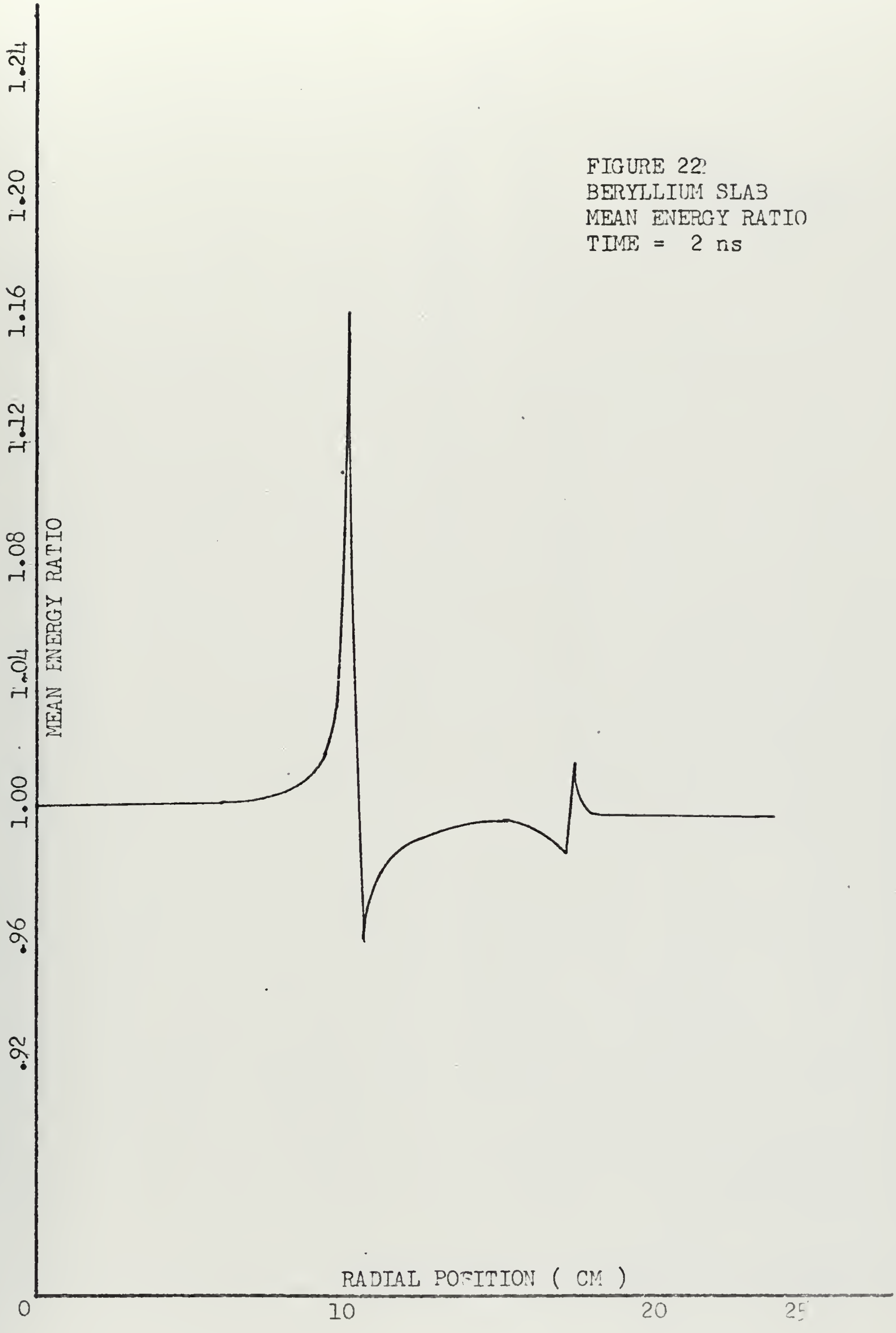


FIGURE 22
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 2 ns
MEAN ENERGY RATIO
1.24
1.20
1.16
1.12
1.08
1.04
1.00
.96
.92
RADIAL POSITION ( CM )
0
10
20
25

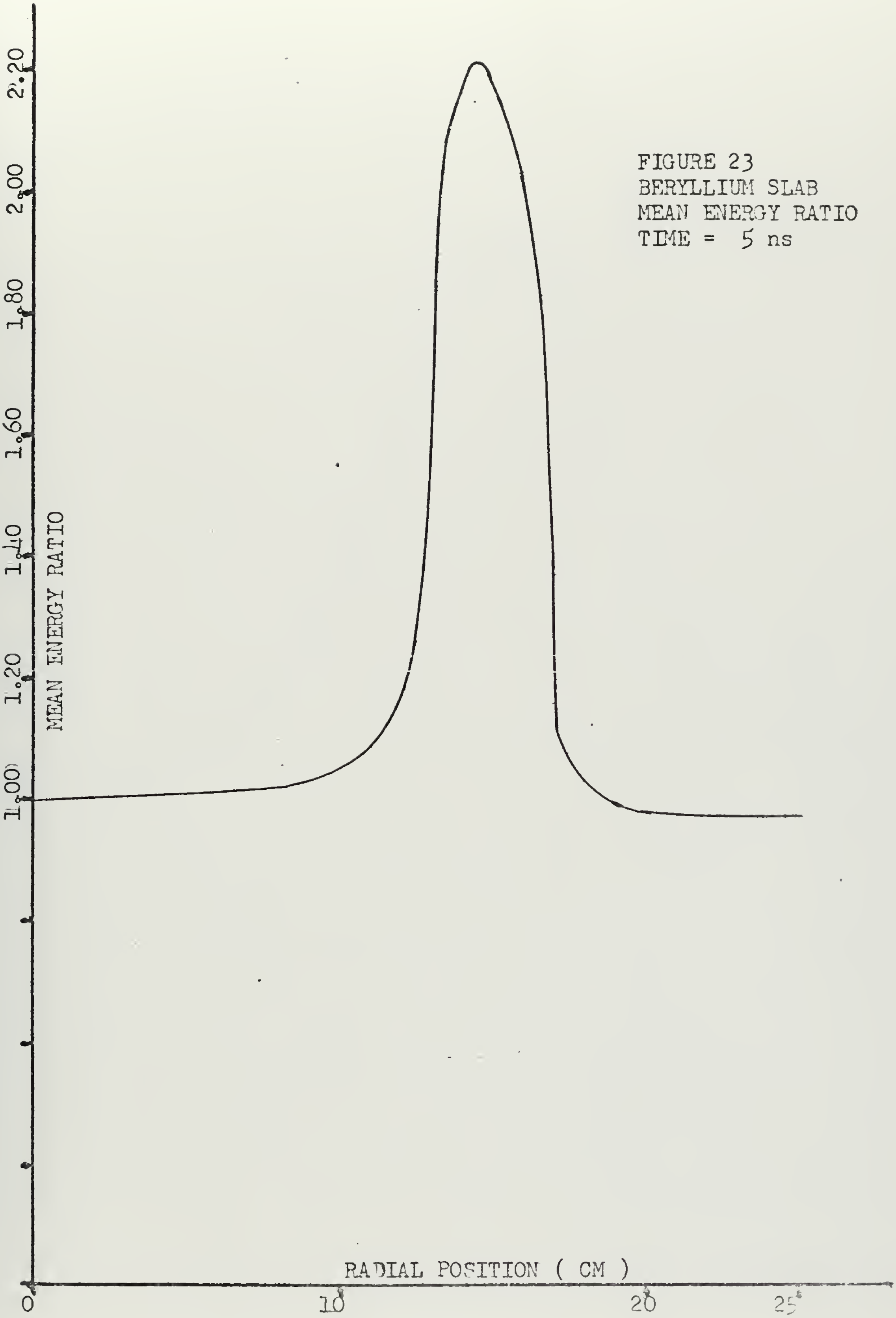


FIGURE 23
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 5 ns
MEAN ENERGY RATIO
RADIAL POSITION ( CM )
2.20
2.00
1.80
1.60
1.40
1.20
1.00
0
10
20
25

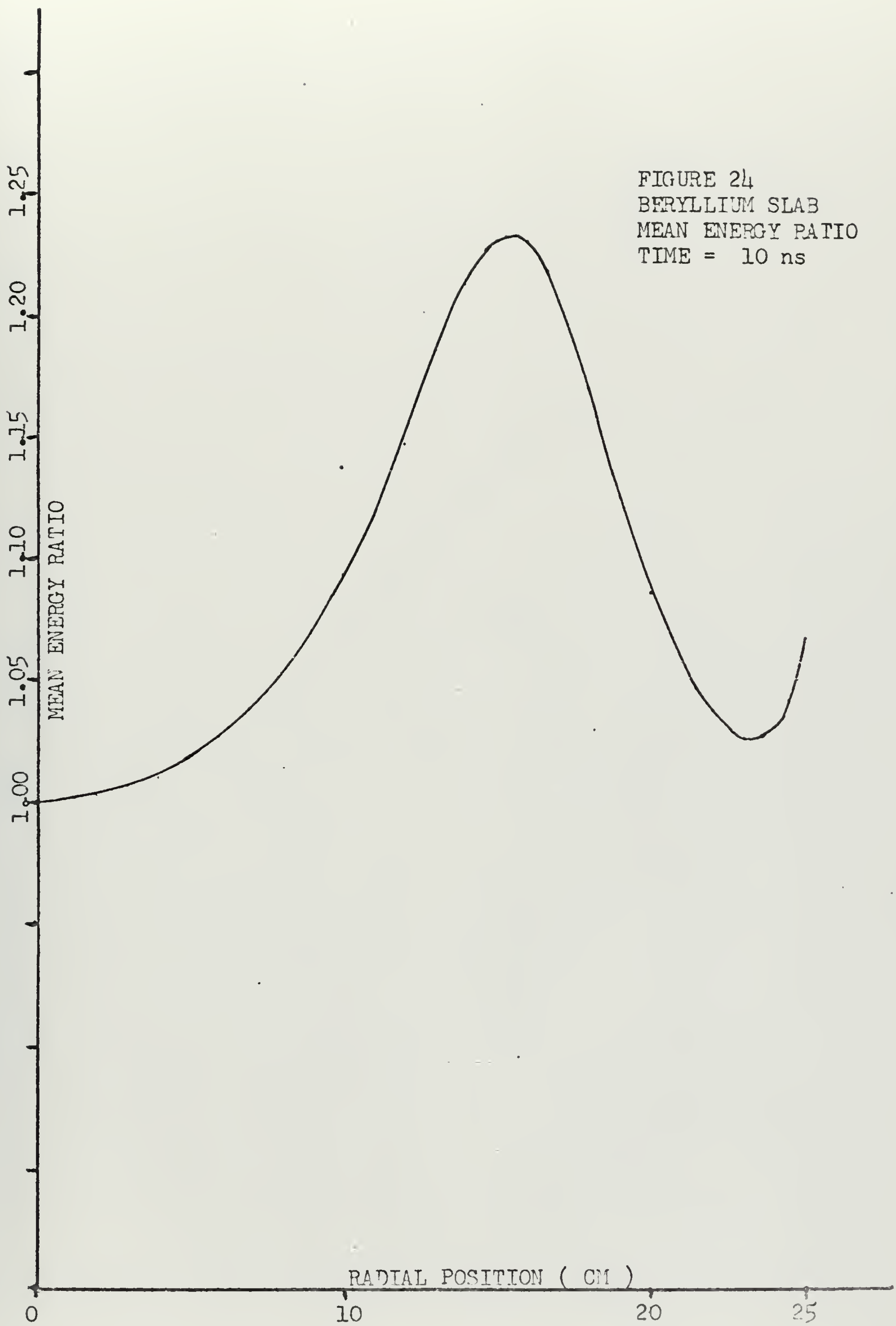


FIGURE 24
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 10 ns
MEAN ENERGY RATIO
1.00
1.05
1.10
1.15
1.20
1.25
RADIAL POSITION ( CM )
0
10
20
25

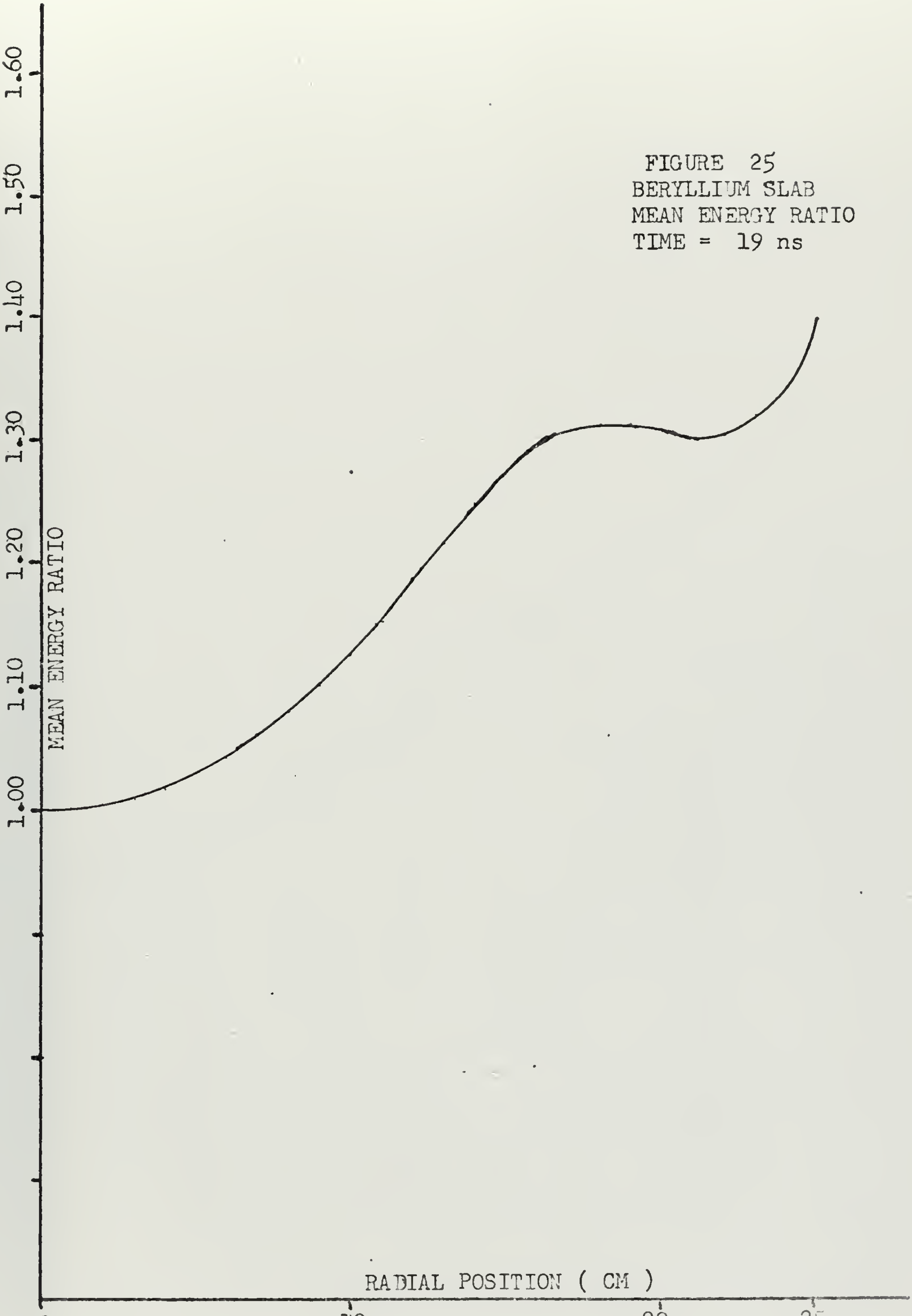



FIGURE 25
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 19 ns

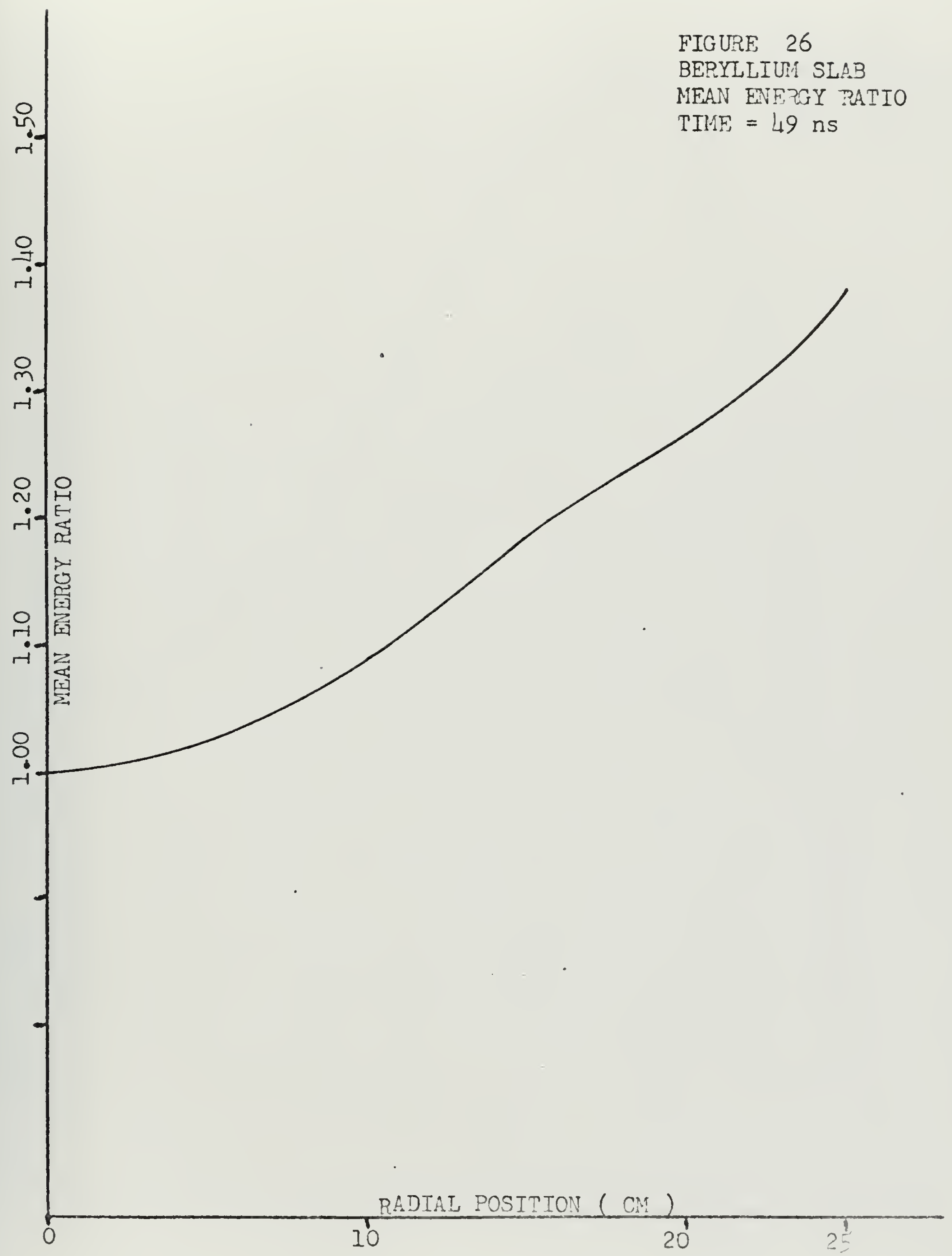


FIGURE 26
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 49 ns
MEAN ENERGY RATIO
1.00
1.10
1.20
1.30
1.40
1.50
RADIAL POSITION ( CM )
0
10
20
25

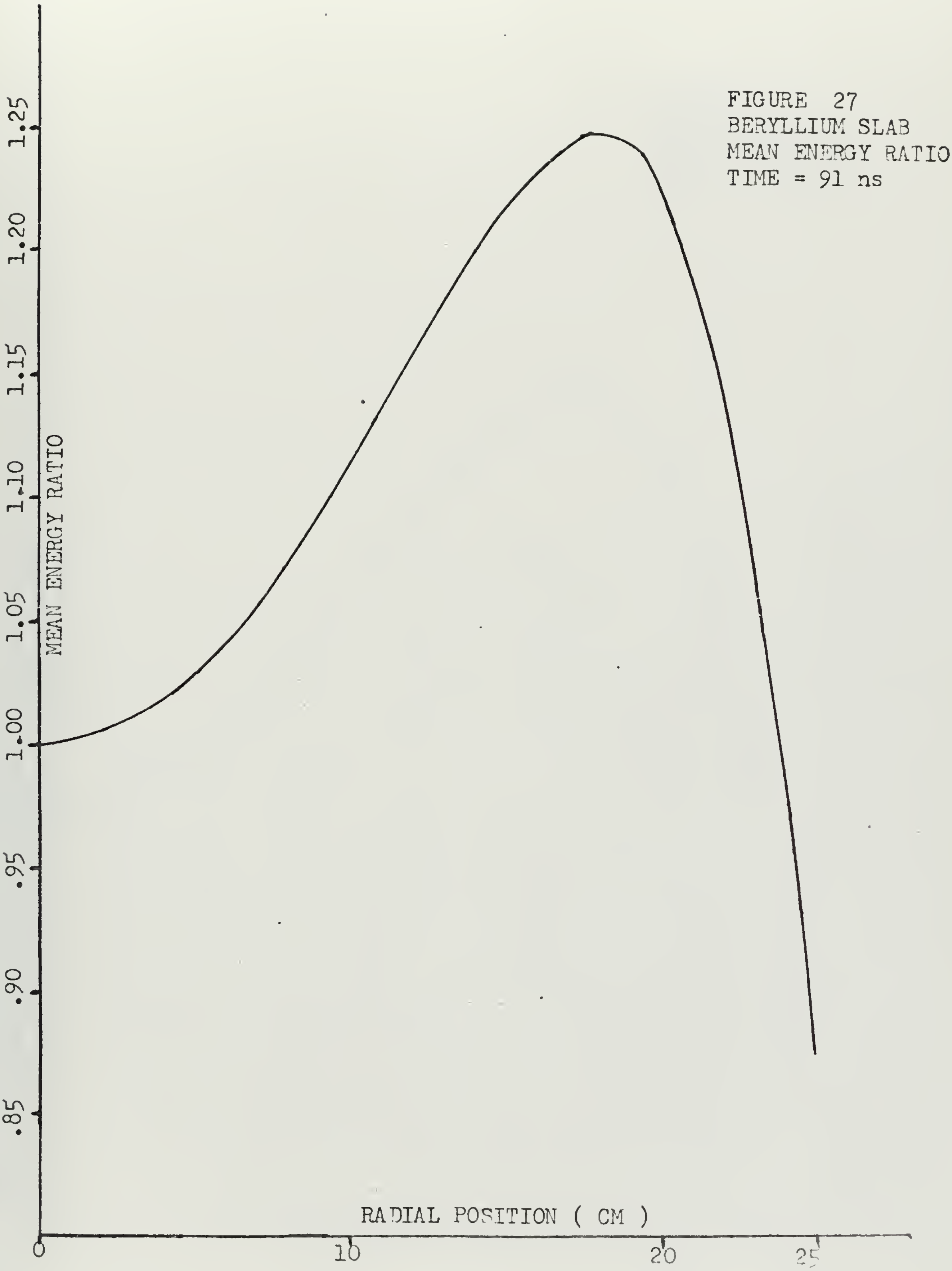


FIGURE 27
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 91 ns
MEAN ENERGY RATIO
RADIAL POSITION ( CM )
1.25
1.20
1.15
1.10
1.05
1.00
.95
.90
.85
0
10
20
25

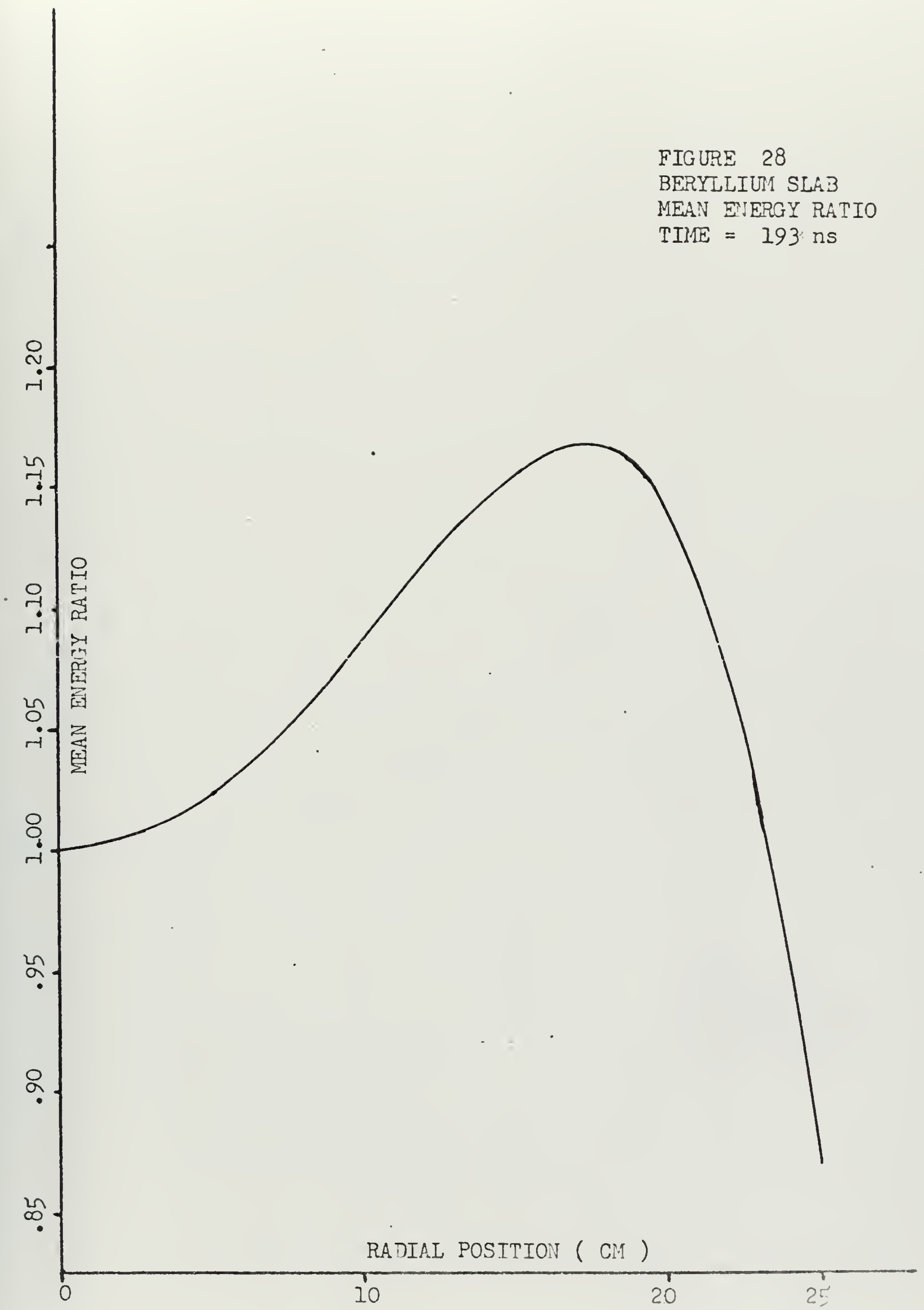


FIGURE 28
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 193 ns
MEAN ENERGY RATIO
1.20
1.15
1.10
1.05
1.00
.95
.90
.85
RADIAL POSITION ( CM )
0
10
20
25

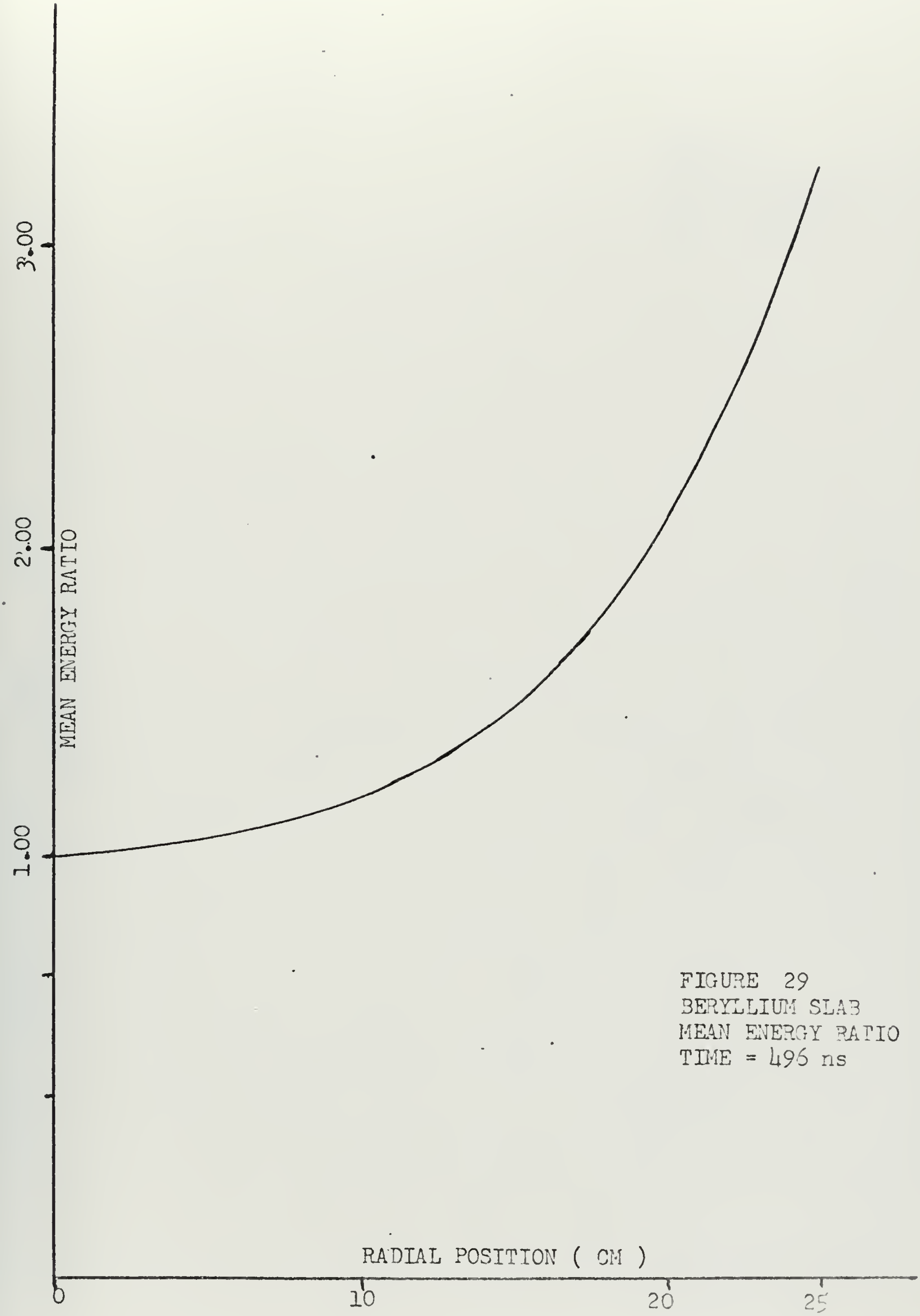



FIGURE 29
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 496 ns

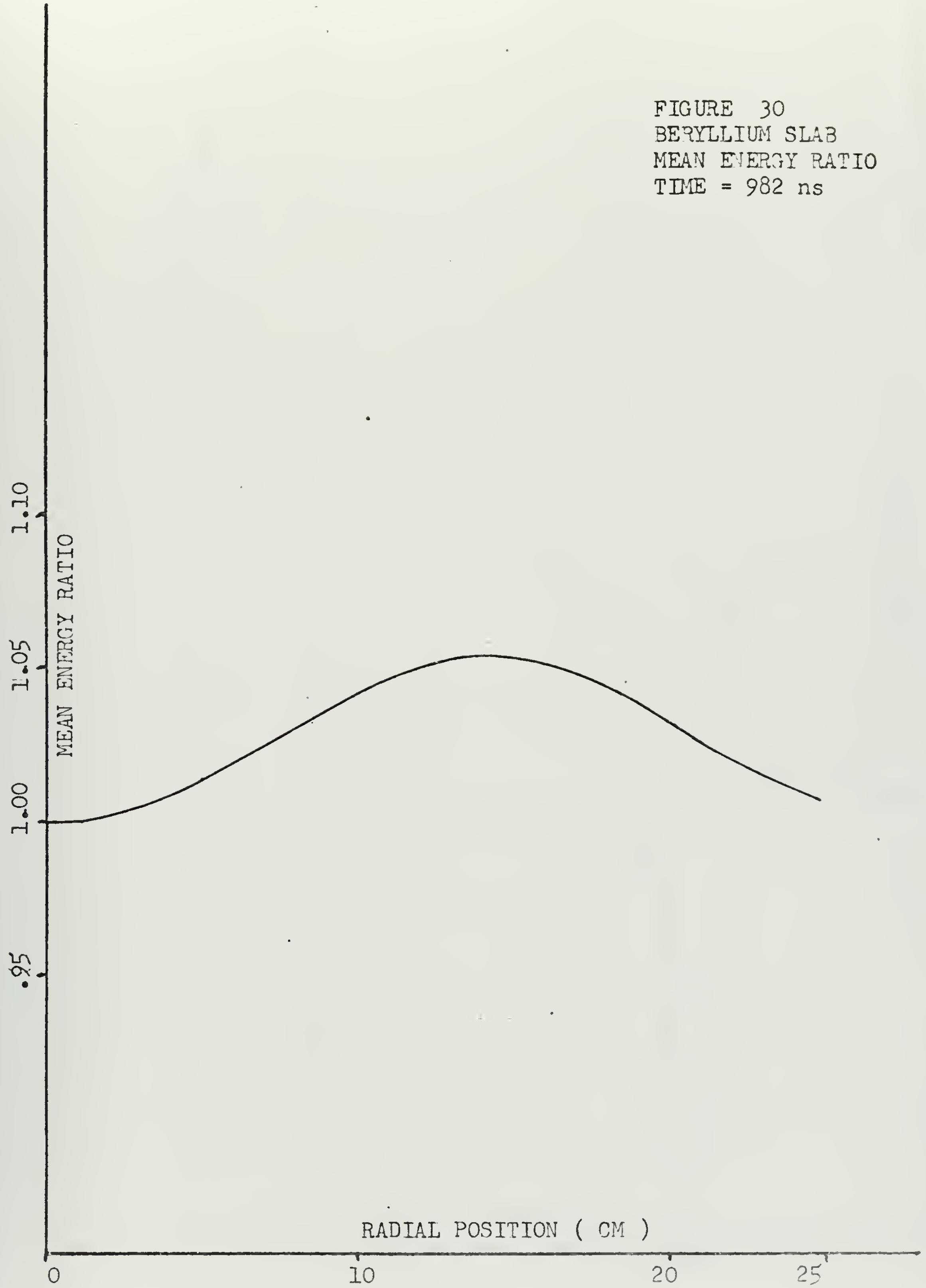


FIGURE 30
BERYLLIUM SLAB
MEAN ENERGY RATIO
TIME = 982 ns
MEAN ENERGY RATIO
1.10
1.05
1.00
.95
RADIAL POSITION ( CM )
0
10
20
25

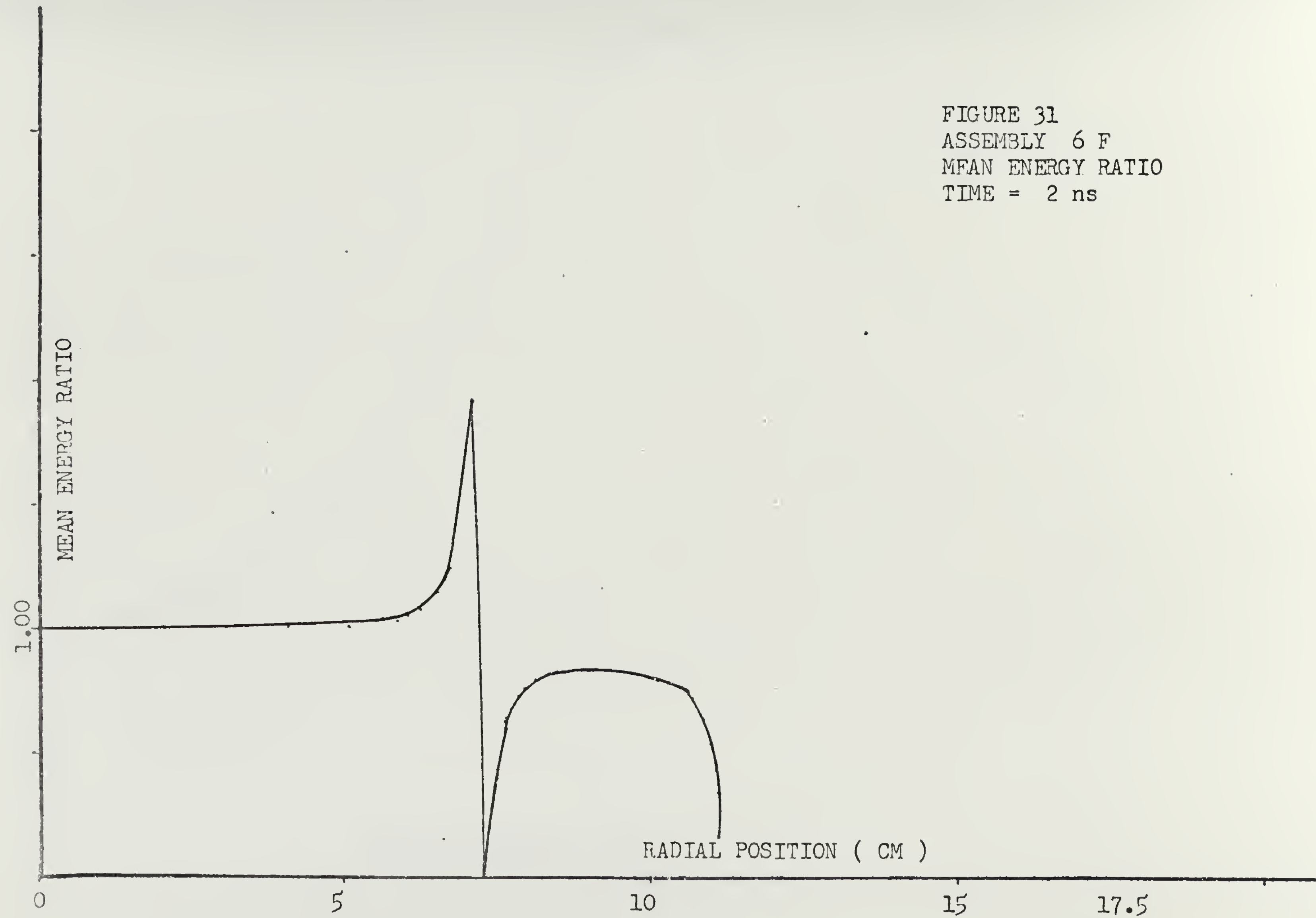



FIGURE 31
ASSEMBLY 6 F
MFAN ENERGY RATIO
TIME = 2 ns

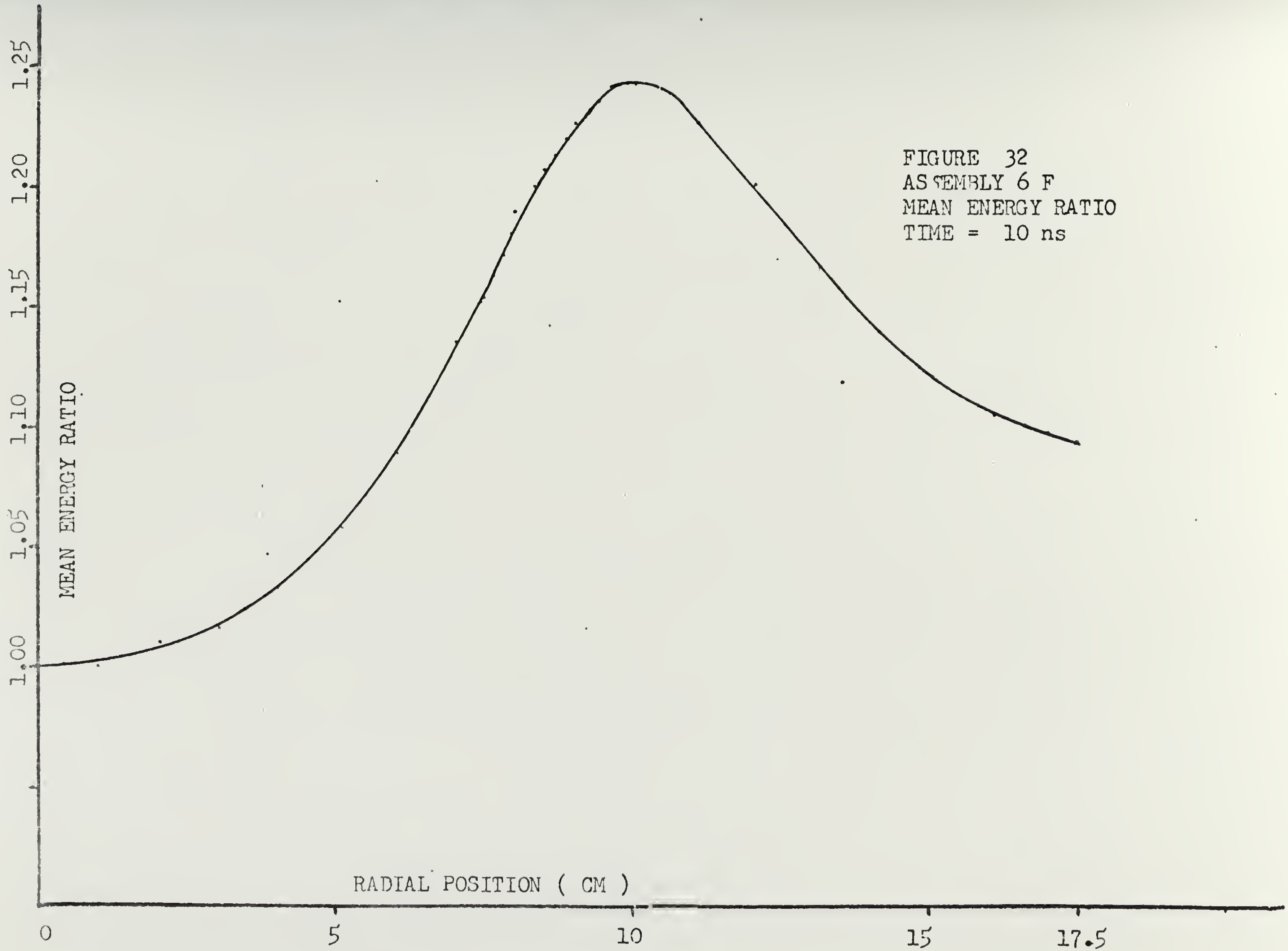



FIGURE 32
ASSEMBLY 6 F
MEAN ENERGY RATIO
TIME = 10 ns

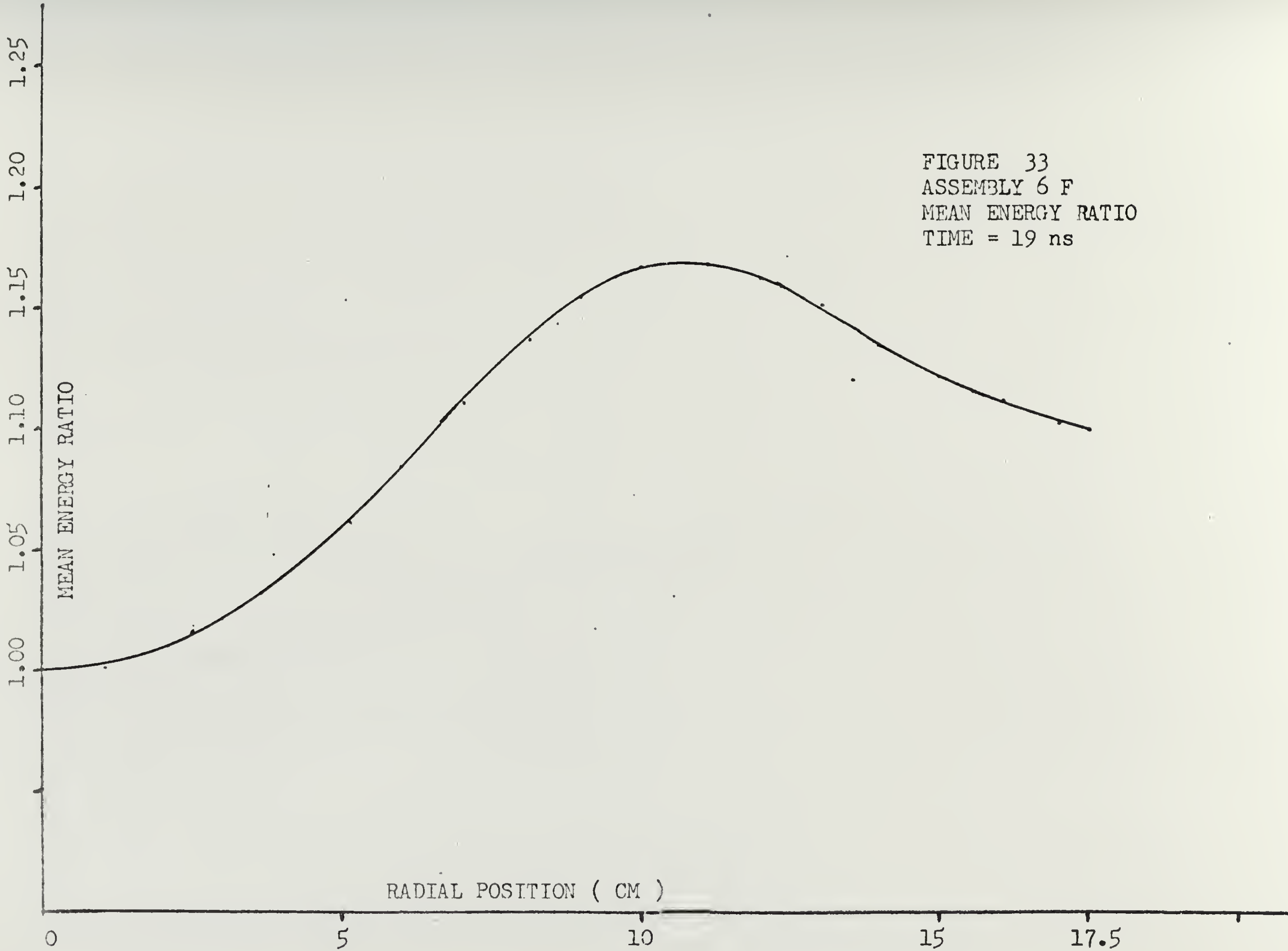



FIGURE 33
ASSEMBLY 6 F
MEAN ENERGY RATIO
TIME = 19 ns

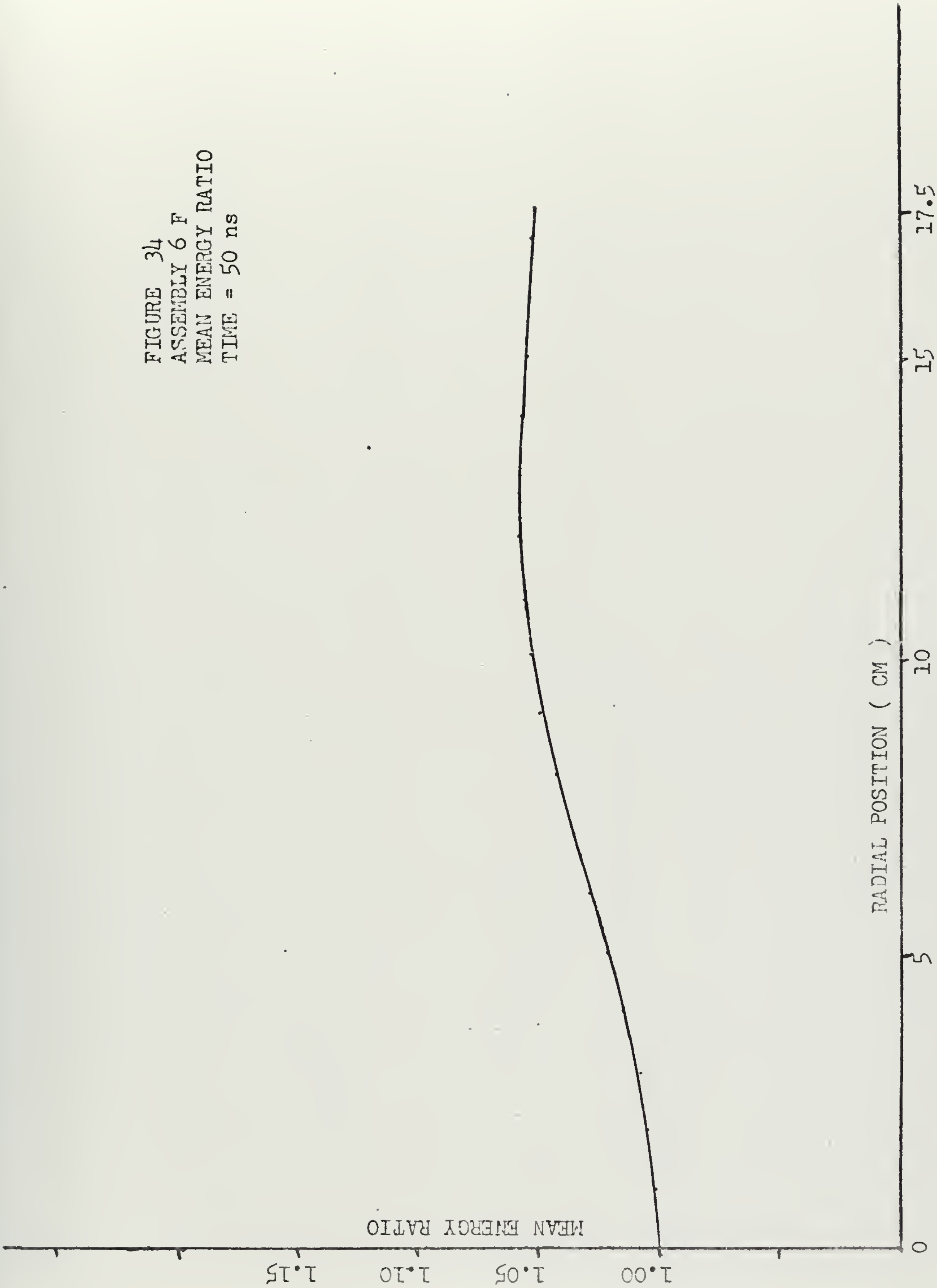
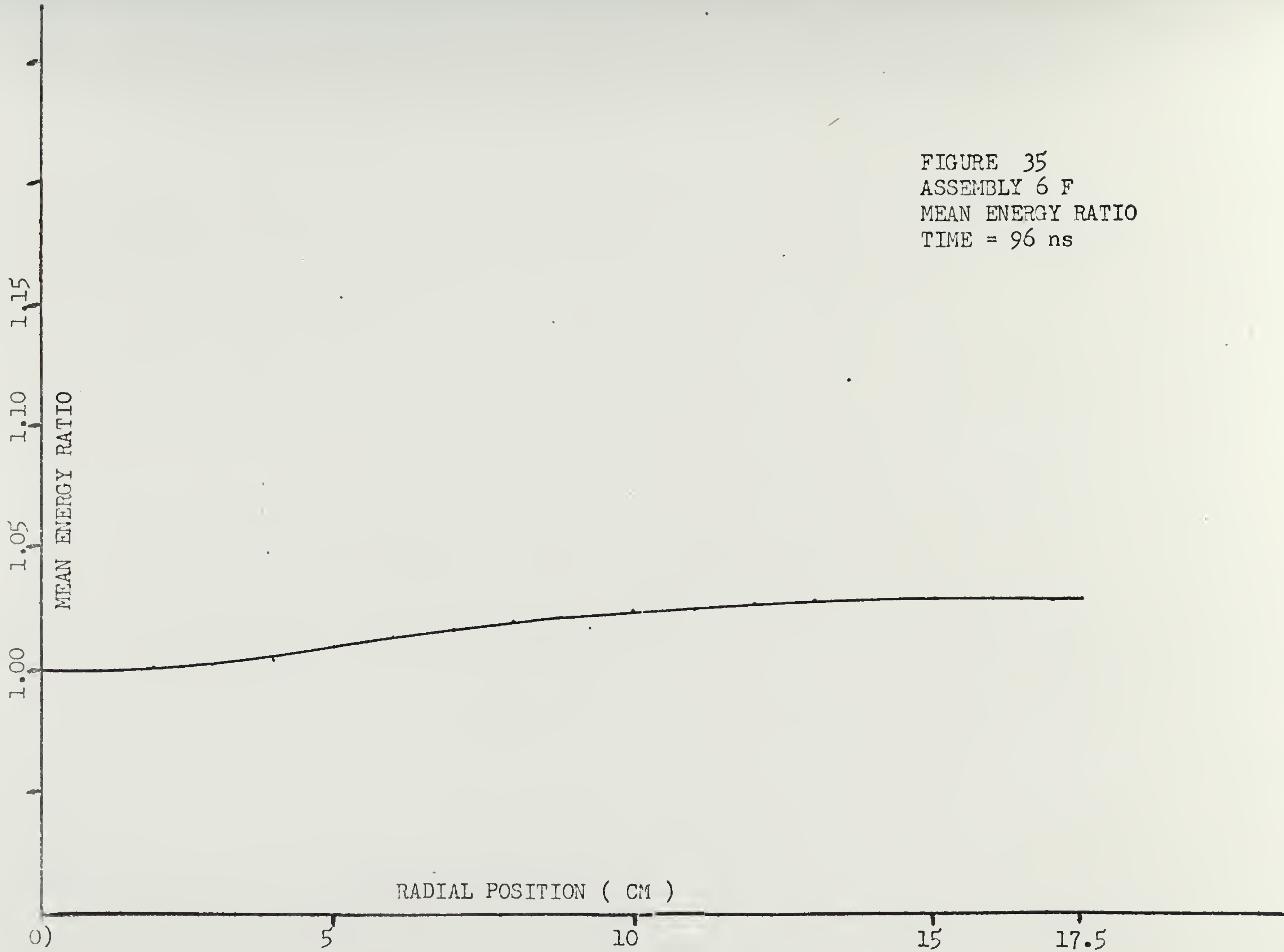



FIGURE 34
ASSEMBLY 6 F
MEAN ENERGY RATIO
TIME = 50 ns

FIGURE 35
ASSEMBLY 6 F
MEAN ENERGY RATIO
TIME = 96 ns



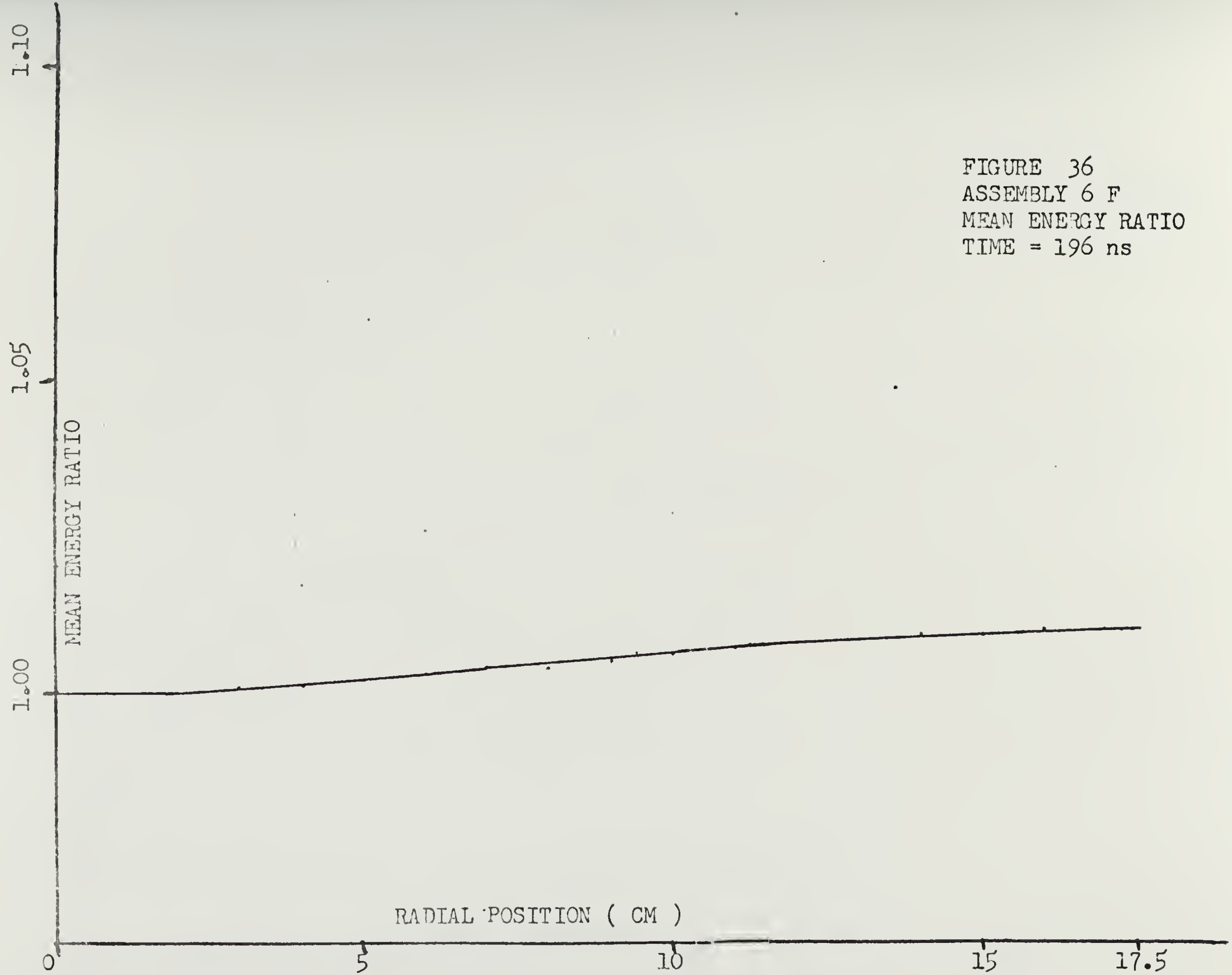



FIGURE 36
ASSEMBLY 6 F
MEAN ENERGY RATIO
TIME = 196 ns

FIGURE 37
BERYLLIUM SLAB
DETECTOR RESPONSE
X = 12.5 CM

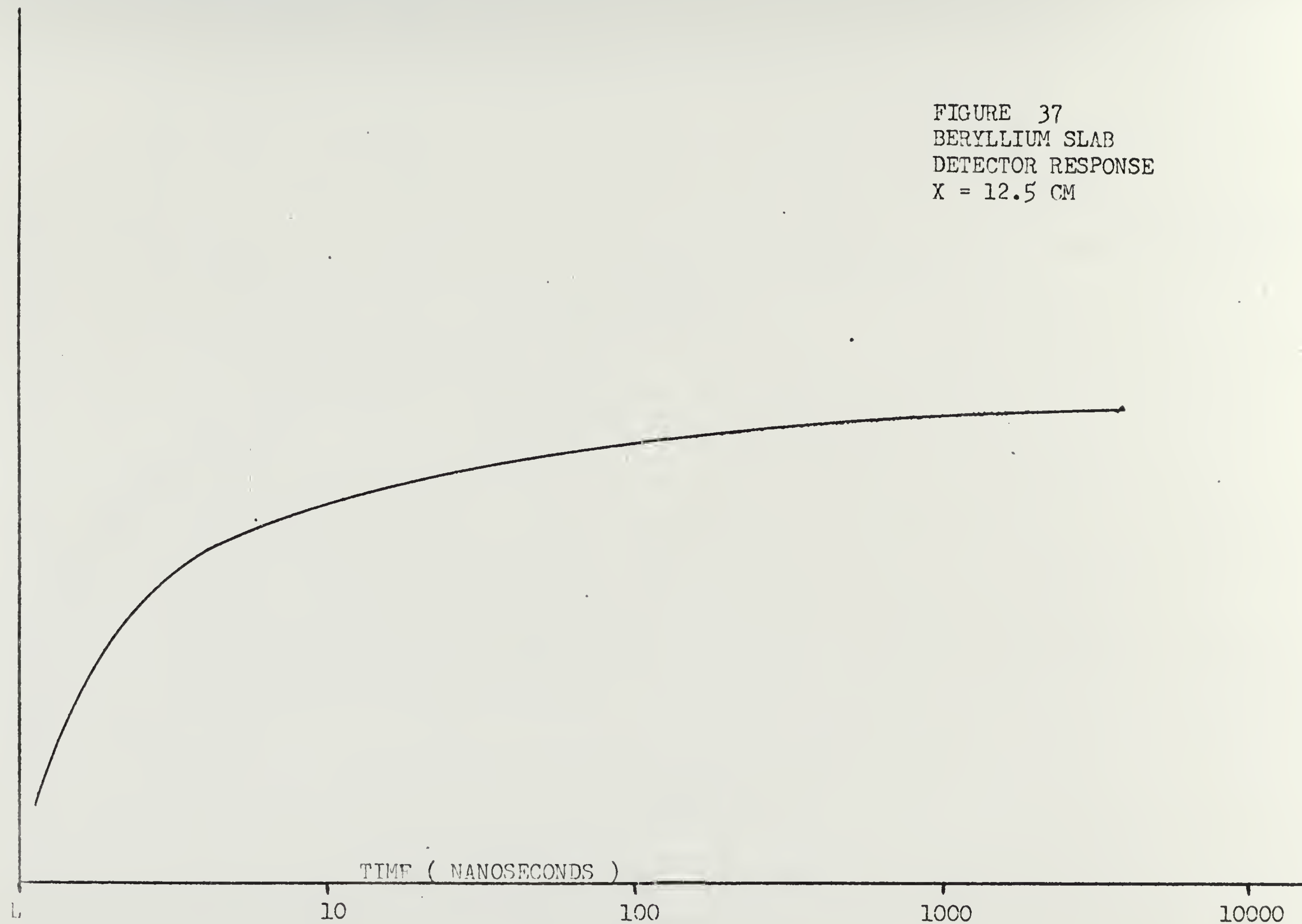



FIGURE 38
ASSEMBLY 6 F
DETECTOR RESPONSE
X = 10.0 CM

```
C
C        REVISED  2  JUNE  1971
C
C     *****************************************************************
C     ***                                                           ***
C     ***   M           M   I   L             SSSSS    I  X     X   ***
C     ***   MM         MM   I   L            S     S   I   X   X    ***
C     ***   M M       M M   I   L     +++    S         I    X X     ***
C     ***   M  M     M  M   I   L     +++     SSSSS    I     X      ***
C     ***   M   M   M   M   I   L     +++          S   I    X X     ***
C     ***   M    M M    M   I   L            S     S   I   X   X    ***
C     ***   M     M     M   I   LLLLLL        SSSSS    I  X     X   ***
C     ***                                                           ***
C     *****************************************************************
C        PROGRAM TO CALCULATE THE POSITIONAL RESPONSE
C        OF FAST NEUTRON INTERACTIONS USING THE OUTPUT OF
C        WILLIAMSON PROGRAM -- MOD - 5
C
C     MAIN PROGRAM
C        PROBLEM INPUT DATA AND CONTROLS
C        FIRST THREE CARDS DEFINE PROBLEM NAME FOR HEADINGS
C        NAMELIST DATA IS INPUT TO DESCRIBE SYSTEM VARIABLES
C        ONE CARDS FOLLOWS WITH SYSTEM CONTROL VARIABLES
C        ONE CARD WITH 20 LOGICAL VARIABLES
C
C        EXAMPLE DATA SET
C        PROBLEM - MIL - NNN - NNN
C        BE CUBE
C        SOURCE TYPE - NAME
C        NAME LIST DATA  &DATAIN
C        &DATAIN IS=  ,SZERO=  ,SOH=  ,SA=  ,S1=  ,S2=  ,S3=  ,S4=  8
C        EMAX=   ,EMIN=   ,
C        IDATA=  ,NOUT=  ,IBX=  ,XBX=  100*A.B,MODES=,NSTAT=  ,&END
C        NANS(IB),IB=1,6     FORMAT (6I2)
C        LOGICAL VARIABLES -- L(I),I=1,20 --<FORMAT> 20I2
C     DEFINITIONS OF PROGRAM VARIABLES
C        CODE INSTRUCTIONS   IDATA =    1    CARD INPUT
C                                       2   TAPE INPUT
C                                       3   DISC INPUT
C        NOUT = 1  PRINT OUTPUT
C        NOUT = 2  PLOT ON THE PRINTER USING ROUTINE -- PLOTP
C        NOUT = 3   CALL CAL COMP  PLOTTER
C        IS = SOURCE TYPE 1 = ON AXIS SOURCE -DEFINE SOH & SA
C                         2 = OFF AXIS SOURCE -- DEFINE SOH,SA,S1
C                         3 = FIRST COLLISION SOURCE --DEFINE SOH,DIFLEN
C                         4 = OFF-AXIS FIRST COLLISION SOURCE
C                             DEFINE SOH,DIFLEN,SA,S1
C                         5 = EXTERIOR SOURCE - RAMP FUNCTION
```

```
C       SOH =  HALF WIDTH OF THE SLAB
C       SA  =  HALF WIDTH OF SOURCE
C       S1  =  MID POINT OF OFF AXIS SOURCE
C       S3  =  MID POINT LOCATION OF DETECTOR
C       S4  =  HALF WIDTH OF DETECTOR
C       DIFLEN = 1.0 / SIGMAT - INVERSE TOTAL MACROSCOPIC CROSS SECTION
C          MODES=  NN          TOTAL MODES FOR PROGRAM TO FOLD IN
C          NSTAT=   M      2        DEFINES DELTA FUNCTION IN LETHARGY
C                                   STATE NUMBER 2
C                         3    DEFINES FISSION SPECTRUM SOURCE
C          IBX = NUMBER OF POINTS ON X AXIS TO CALCULATE DATA ABOUT
C          XBX(NN) INPUTS THE POSITIONS  TO DO CALCULATIONS FOR
C                    TO A MAXIMUM OF 100 DATA POINTS ALONG THE X-AXIS
C       EMAX - MAXIMUM ENERGY OF NEUTRON THAT A DETECTOR CAN MEASURE
C       EMIN - MINIMUM ENERGY THAT THE DETECTOR CAN MEASURE
C
C          USE OF THE LOGICAL CONTROL VARIABLES
C          L( 4)     EXTERNAL CONTROL - IN SUBROUTINE FOUT5 SPECTRAL
C                    RESPONSE-PLOT THE SPECTRA AT POINT OTHER THAN X=0.0
C                    USE INPUT VARIABLES NANS(6) FOR TOTAL POINTS TO DO.
C                    USE POSITION VARIABLES XBX(1) TO XBX(NN)
C       L(5) =   1   EXTERNAL CONTROL -BYPASS THE PRINT SEQUENCE IN FOUT2
C       L(7) =   1   EXTERNAL CONTROL - PLOT FLUX PROFILE VS X ON CALCOMP
C       L(8) =   1   PLOT MEAN ENERGY RATIO VS X ON CAL COMP IN FOUT2
C          L( 9)     EXTERNAL CONTROL -PERFORM SIMPLE PLOT OF SOURCE
C                    FUNCTION ON THE PRINTER IN ROUTINE MODEL1
C          L(10)     INTERNAL CONTROL - TERMINATE RUN AFTER INCON1
C                    IF ERROR IN PROBLEM DEFINITION OR CONTROL MODES
C          L(14)     CALL FOUT2-CALC MEAN ENERGY-E(X,T) AND NEW F(X,T)
C          L(16)     INTERNAL CONTROL - BY PASS OUTPUT ROUTINES IN
C                    SUBROUTINE FOUT1 ON CALLS FROM ANY OTHER ROUTINE IN
C                    THE REMAINDER OF THE PROGRAM
C          L(17)     CALL SPECTRAL PLOT ROUTINE FOUT5
C          L(18)     CALL FOUT1 FOR INPUT DATA POINTS
C          L(19)     CALL SUBROUTINE FOUT3 - DETECTOR RESPONSE
C          L(20)     CALL SUBROUTINE FOUT4 FOR SPECTRAL PLOT WITHOUT
C                    ADJUSTMENT BY THE FOURIER COEFFICIENT
C          PROGRAM OUTPUT CONTROL VARIABLES
C          NANS( 1 ) -EXTERNAL CONTROL - LIMIT TO TOTAL GRAPHS TO BE
C                     PLOTTED IN SUBROUTINE FOUT1   DEFAULT OF 1 GRAPH
C          NANS( 2 ) -INTERNAL CONTROL - LABELING OF FOUT1 GRAPHS ON
C                     CALLS FROM MAIN PROGRAM AND IN ROUTINE FOUT2
C          NANS( 3 ) -EXTERNAL CONTROL - LIMIT TO TOTAL GRAPHS TO BE
C                     PLOTTED BY FOUT2    DEFAULT OPTION = 1
C          NANS( 4 ) - EXTERNAL CONTROL - LIMIT TO ALLOWABLE NUMBER
C                       OF INTERATIONS IN ROUTINE MODEL1
C          NANS( 5 ) -NOT USED     10 MAY
C          NANS( 6 ) NUMBER OF POINTS FOR SPECTRAL RESPONSE PLOTTING
```

```
C                       WILL ONLY BE CONSIDERED FOR CASES WITH L( 4 )= 1
C
C
C        MODIFICATION OF MIL SIX  MOD 2
C        COMMON BLOCKS 3 AND 6  MUST BE REVISED TO HANDLE MORE THAN
C        SIX HARMONIC MODES OF THE FUNCTIONAL DATA INPUT
         COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
         COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
         COMMON/COM7/ SO1,SO2,SO3,SO4,AK11,AK12,AK13,AK14,AN3A,AN3B,
        1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
         COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
         COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
        1NSTAT,MODE
         COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
         COMMON/COM6/ NTOP(6,20),NM(6,20),IKNT(6)
C
         CALL CANCEL(2)
C
C        SET INITIAL CONDITIONS OF THE PROGRAM
       1 CALL INCON1
C        INTERNAL PROTECTION IF INPUT ERROR IN PROPLEM DEFINITION
         IF (L(10).EQ.1) GO TO 52
       2 IF(IDATA.EQ.0) GO TO 60
         IF (IDATA.EQ.1) CALL READ10(LASTN)
         IF (IDATA.EQ.2) CALL READ11(LASTN)
C
C           2    SUBROUTINE ESTIM1          INITIAL PROGRAM ESTIMATES OF
C                       TOTAL PROGRAM RUNNING-EXECTUION TIME DURING TESTING
C        PERFORM INITIAL TIME ESTIMATING OF THE PROGRAM EXECUTION TIME
C        REQUIREMENTS WHILE STILL DOING INITIAL DEBUGGING AND CHECKOUT
C
         CALL ESTIM1
C
         CALL MODEL1
C
C        INITIAL DEFINITION AND ERROR ANALYSIS BY ROUTINE MODEL1
         IF (LASTN.EQ.1) GO TO 50
C        THIS WILL BY PASS ALL FOLLOWING STEPS IF THERE IS AN INPUT ERROR
       6 CONTINUE
C        PARAMETER L(20) = 1 WILL CALL FOR A PLOT OF THE INPUT SPECTRUM
C        OF THE MOD-5 DATA WITH A PLOT OF THE DENSITY VECTOR
C        VERSUS THE LETHARGY VALUES
         IF (L(20).EQ.1) CALL FOUT4
C
C        FOUT5 WILL PLOT SPECTRUM FOR EACH MODE WEIGHTED BY HARMONIC
C        FOURIER COEFFICIENT FOR EACH MODE AND THE SUMMED SPECTRUM
C        FOR ALL MODES OF INPUT DATA
         IF (L(17).EQ.1) CALL FOUT5
      20 CONTINUE
```

```
   30 CONTINUE
C
C     LOGICAL PARAMETER L(18) = 1 RESULTS IN THE POSITION FUNCTION
C     F(U,T,X) BEING CALCULATED FOR THE POSITIONS  XBX(I)
C     AS STATED IN THE INPUT DATA
   38 IF (L(18).EQ.0) GO TO 65
C
C     CALLING ROUTINE TO CALCULATE THE FUNCTION--F(U,T,X) FOR POSITIONS
C     DESIRED AS LISTED IN THE INPUT DATA FOR THAT POINT
C     IF ANY OUTPUT IS DESIRED FROM ROUTINE FOUT1 VIA INITIAL
C     CALLING SEQUENCE FROM THE MAIN PROGRAM -- DEFINE L( 16 ) = 0
C     IN THE PROBLEM INPUT STRUCTURE
   40 IF (IBX.EQ.0 ) XBX(1) = 0.
      IF ( IBX.EQ.0) IBX=1
      IF (IS.EQ.0) IS=1
C     PROTECTION STEP TO ALLOW FOUT1 TO PRINT OUT THE VALUES
C     OF THE FUNCTION  --  F(U,T,X)  ON THE CALLS FROM MAIN PROGRAM
      L(16) = 0
      DO 43 I1X = 1,IBX
      IF (NOUT.EQ.3) NANS(2) = I1X
      WRITE (6,63)
      CALL FOUT1(XBX(I1X))
   43 CONTINUE
   65 CONTINUE
      IF (L(14).EQ.1) CALL FOUT2
C
C        1    SUBROUTINE FOUT2           CALC MEAN ENERGY AND PLOT SUM
C                  OVER ALL LETHARGY STATES OF F(U,T,X) AND PLOT
C                  BOTH FUNCTIONS ON THE SAME PLOT FOR EACH TIME STEP
C
      IF(L(19).EQ.0) GO TO 48
C     PARAMETER L(19) = 1 AS DEFINED IN THE PROBLEM DEFINITION
C     TO CALCULATE THE DETECTOR RESPONSE IN SUBROUTINE FOUT3
C     IF L(19) = 0   --- WE WILL BY-PASS THIS ROUTINE
C
      WRITE (6,64)
      CALL FOUT3
   48 CONTINUE
   49 CONTINUE
      GO TO 60
C
C     THIS WILL ALLOW A BYPASS AND STOP IF ERROR
   50 WRITE (6,51)
      GO TO 60
   52 WRITE (6,53)
   60 CONTINUE
      WRITE (6,61) (NPROB(IA),IA=1,54)
   51 FORMAT (///,10X,'INPUT ERROR IN DATA FILE - F(U,T) FROM MOD5',/,
```

```
 110X,'PROGRAM TERMINATED')
53 FORMAT (///,'ERROR IN PROBLEM STRUCTURING',/,
 110X,'RUN TERMINATED')
61 FORMAT (///,3(10X,18A4,/),10X,4(3X,'***END***',3X))
63 FORMAT ('1',10X,'EXECUTION BEGINS WITH CALL TO FOUT1')
64 FORMAT ('1',10X,'EXECUTION CONTINUES WITH CALL TO FOUT3')
   STOP
   END


   SUBROUTINE INCON1
C     REVISED  2  JUNE  1971
C
C     THIS ROUTINE DOES THE FOLLOWING
C  1  SETS ALL ARRAY LOCATIONS TO ZERO
C  2  DEFINES ALL PRINCIPAL DEFAULT OPTIONS,I.E.
C     SOH    =  100.0  CM
C     SA     =    1.0  CM
C     S1     =    0.0  CM
C     S2  S3  S4  = 0.0
C     DIFLEN = 25.0
C     IS     =    1
C     IDATA  =    1
C     MODES  =    1
C     NSTAT  =    2
C     IBX=0
C     SZERO  =    1.0
C  3  READS ALL PROBLEM DEFINITION STATEMENTS FROM PUNCH CARDS
C     AND ECHO PRINTS ALL INPUT DATA
C  4  INFORMS THE USER WHAT ACTIONS THE PROGRAM WILL EXECUTE
   REAL LTAB(20)/'T1','T2','T3','T4','T5','T6','T7','T8','T9',
  1'T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'/
   DIMENSION NDATAC(18)
   NAMELIST/DATAIN/ IS,SZERO,SOH,SA,S1,S2,S3,S4,IDATA,NOUT,IBX,XBX,
  1DIFLEN,MODES,NSTAT,EMAX,EMIN
C  WITHIN INCON1 WE WILL HAVE TO ESTABLISH SOME FORM OF DEFAULT
C  OPTION FOR THE MAX AND MIN ENERGYS THAT THE DETECTOR RESPONSE
C  ROUTINE WILL HAVE TO CONSIDER
   COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
   COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
   COMMON/COM7/ S01,S02,S03,S04,AK11,AK12,AK13,AK14,AN3A,AN3B,
  1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
   COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB
   COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
  1NSTAT,MODE
   COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
   COMMON/COM6/ NTOP(6,20),NM(6,20),IKNT(6)
C
```

```
C       SET INITIAL DEFAULT VALUES
C       INITIALLY DEFINE ALL PARAMETERS TO ZERO VALUES
        CALL CANCEL(2)
        DO 2 I=1,150
        U(I) = 0.0
        E(I) = 0.0
        DUMMY(I) = 0.0
        DO 1 J= 1,20
      1 FUTX(I,J) = 0.
      2 CONTINUE
        DO 4 I= 1,100
        DO 3 J=1,20
      3 FXT(I,J) = 0.0
      4 TXBX(I) = 0.0
        DO 41 I=1,20
        NT(I) = 0
        TIM(I) = 0.0
        DO 40 J=1,6
        NTOP(J,I) = 0
        NM(J,I) = 0
     40 CONTINUE
     41 CONTINUE
        DO 42 I=1,6
        DO 42 J=1,20
        DO 42 K=1,71
     42 FUT(I,K,J) = 0.0
        DO 43 J=1,6
     43 BUCK(J) = 0.0
C       ESTABLISHMENT OF PRINCIPAL DEFAULT VALUES FOR THE FIRST TESTING
C       OF THIS ROUTINE
C
C       DEFAULT TO PROCESS ONLY ONE DATA SET
        DIFLEN = 25.0
        SOH = 100.
        SA   =  1.
        S1 = 0.
        S2 = 0.
        S3 = 0.
        S4 = 0.
        IBX=0
        IS = 1
        IDATA = 1
        SZERO = 1.
        XBX(1)=0.0
        MODES = 1
        NSTAT = 2
C
C       DEBUG DATA LIST --- DUPLICATE OF ENTRY CARDS
```

```
      WRITE (6,28)
      READ (5,29) NCARD
      DO 27 IIM=1,NCARD
      READ (5,30) (NDATAC(IA),IA=1,18)
      WRITE (6,31) (NDATAC(IA),IA=1,18)
   27 CONTINUE
C     END OF DEBUG DATA PRINT OUT
C     START BY READING THE PROBLEM TITLE
      READ (5,9) (NPROB(IA),IA=1,54)
C     READ IN THE NAMELIST DATA
      READ (5,DATAIN)
C     INPUT OF PROGRAM LOGICAL CONTROLS
      READ (5,11) (NANS(I),I=1,6)
      READ (5,12) (L(I),I=1,20)
C     DUPLICATE DATA PRINT OUT OF THE PROBLEM DEFINITION CARDS
C     OUTPUT OF PROBLEM STATE STRUCTURE
C     START PROBLEM LISTING ON A NEW PAGE
      WRITE (6,28)
      WRITE (6,60) (NPROB(IA),IA=1,54)
      WRITE (6,61) IS,IDATA,NOUT,IBX,SZERO,DIFLEN,SOH,SA,S1,S2,S3,S4
      WRITE (6,63) MODES,NSTAT
      IF (IBX.EQ.0) GO TO 68
C
C     PROTECTION STEPS TO TERMINATE PROGRAM EARLY IN EXECUTION IF
C     SOME ERRORS EXIST IN THE BASIC PROBLEM STATEMENT
C     COMPARE ALL PERTAINENT DATA DEPENDING ON THE SUBROUTINES TO
C     BE DONE TO INSURE THAT ALL NECESSARY INPUT HAS BEEN PROVIDED
C     LOGICAL L( 10 ) = 0  INITIALLY-- IF INPUT DATA IS IN ERROR,
C     L( 10 ) SET = TO 1  PROBLEM WILL BE TERMINATED IN MAIN PROGRAM
C     ON RETURN FROM THIS SUBROUTINE.
      L(10)=0
      IF ((IBX.LT.0).OR.(IBX.GT.100)) L(10)=1
      IF (IBX.GT.100) IBX=100
      IF (IBX.GT.0) WRITE (6,62) (XBX(J),J=1,IBX)
      GO TO 69
   68 WRITE (6,67) XBX(1)
   69 WRITE (6,70) (NANS(IA),IA=1,6)
      WRITE (6,72) (L(I),I=1,20)
      IBX=1
C
C     COMPLETE TEST SERIES TO EXAMINE MAIN INPUT CONTRAL DATA
      IF ((NOUT.GT.3).OR.(NOUT.LE.0)) L(10)=1
      IF ((IS.GT.5).OR.(IS.LE.0)) L(10)=1
      IF ((SOH.LE.SA).OR.(SOH.LE.S1).OR.(SOH.LE.S3)) L(10)=1
      IF ((NSTAT.GE.4).OR.(NSTAT.LE.1)) L(10)=1
      IF ((MODES.LT.1).OR.(MODES.GT.6)) L(10)=1
C     TEST FOR TOTAL MODES MUST BE REVISED IF PROGRAM IS MODIFIED
C     TO PROCESS MORE THAN 6 MODES OF DATA
```

```
C     THIS TEST QUESTION MUST BE CHANGED TO REFLECT THE STORAGE
C     ARRAY CAPABILITIES OF COMMOM BLOCKS   COM3 & COM6
C
C        PROGRAM WILL WRITE THE USER A SERIES OF MESSAGES TO DESCRIBE
C        WHAT ACTIONS THE PROGRAM WILL EXECUTE IN THE PERFORMANCE
C        OF THIS SPECIFIC JOB
  200 WRITE (6,220)
      GO TO ( 201,202,203,204,2055),IS
  201 WRITE (6,221) SOH,SA
      GO TO 2041
  202 WRITE (6,222) S1,SA,SOH
      GO TO 2041
  203 WRITE (6,223) SOH,DIFLEN
      GO TO 2041
  204 WRITE (6,2231) S1,SOH,DIFLEN
      GO TO 2041
 2055 WRITE (6,2250)
 2041 CONTINUE
      GO TO ( 205,206,207),IDATA
  205 WRITE (6,224)
      GO TO 208
  206 WRITE (6,225)
      GO TO 208
  207 WRITE (6,226)
  208 GO TO (209,210,211),NOUT
  209 WRITE (6,227)
      GO TO 212
  210 WRITE (6,228)
      GO TO 212
  211 WRITE (6,229)
  212 CONTINUE
  213 WRITE (6,230) MODES
C     PROGRAM EXECUTION LISTING TO ADVISE THE USER THE ROUTINES
C     THAT WILL BE EXECUTED AND THE PRINCIPAL ITEMS OF DATA THAT
C     WILL BE USED BY EACH OF THEM
C     THIS WILL ASSIST THE USER IN ANY CASES WHERE SOME UNEXPECTED
C     EXECUTION ERROR MAY OCCUR
C     STATEMENTS ARE LISTED IN THE ORDER IN WHICH THEPROGRAM WILL
C     EXECUTE THEM
C     FOUT4 -- SPECTRAL DATA PLOT
  300 IF (L(20).EQ.0) GO TO 301
      WRITE (6,320) MODES
C     FOUT5 -- FOURIER ADJUSTED SPECTRAL DATA PLOT
  301 IF (L(17).EQ.0) GO TO 302
      WRITE (6,321) MODES
      NPXT5=NANS(6)
      IF (L(4).EQ.1) WRITE(6,3211) NANS(6),(XBX(IJ),IJ=1,NPXT5)
C     FOUT1 - CALCULATE F(U,T,X) = F(U,T)*PHI(X) FOR INPUT DATA
```

```
  302 IF (L(18).EQ.0) GO TO 303
      WRITE (6,322) IBX,NOUT,NANS(1),(XBX(IJ),IJ=1,IBX)
C     FOUT2 -- MEAN ENERGY AND FLUX PROFILE
  303 IF (L(14).EQ.0) GO TO 304
      WRITE (6,323) NANS(3)
      IF ( L(5).EQ.1) WRITE (6,3231)
      IF (L(7).EQ.1) WRITE (6,3232)
      IF (L(8).EQ.1) WRITE (6,3233)
C     FOUT3 -- DETECTOR RESPONSE
  304 IF (L(19).EQ.0) GO TO 305
      WRITE (6,324) S3,S4,EMAX,EMIN,NOUT
  305 CONTINUE
    9 FORMAT (18A4)
   11 FORMAT (6I2)
   12 FORMAT (20I2)
   28 FORMAT('1',//,20X,'INPUT DATA FILE')
   29 FORMAT (6X,I4)
   30 FORMAT (18A4)
   31 FORMAT (10X,18A4)
   60 FORMAT (10X,18A4)
   61 FORMAT (10X,'IS = ',6X,I3,/,10X,'IDATA = ',3X,I3,/,10X,'NOUT =',
     14X,I3,/,10X,'X-POINTS =',I4,/,10X,'SOURCE STRENGTH =',2X,F6.2,
     2/,10X,'DIFFUSION LENGTH =',F5.2,/,10X,'SOH = ',F6.2,3X,'SA =',
     32X,F5.2,2X,'S1 = ',F5.2,/,10X,'S2 =',3X,F5.2,2X,'S3 = ',2X,F5.2,
     43X,'S4 =',2X,F5.2)
   62 FORMAT (10(5X,10(F8.3,2X),/))
   63 FORMAT (//,10X,'MODES =',2X,I5,/,10X,'NSTAT =',3X,I5)
   67 FORMAT (10X,F8.5)
   70 FORMAT (//,10X,'LOGICAL CONTROL VARIABLES - NANS(I)',/,
     110X,6(5X,I3))
   72 FORMAT (//,10X,'PROGRAM LOGICAL CONTROL VARIABLES',/,
     16X,4(3X,5I3,3X))
  220 FORMAT ('1',10X,3(2X,'****',2X),//,10X,'MIL SIX FOLDING',
     11X,'PROGRAM',//,11X,3(2X,'****',2X),//,10X,'PROGRAM WILL',
     21X,'EXECUTE THE FOLLOWING')
  221 FORMAT (/,10X,'DELTA FUNCTION SOURCE ON AXIS AT X = 0.0',/,
     1T10,'SLAB HALF THICKNESS = ',1X,F8.2,1X,'SOURCE HALF THICKNESS',
     21X,'=',1X,F6.2)
  222 FORMAT (/,10X,'DELTA FUNCTION SOURCE OFF AXIS -CENTERED AT X =',
     12X,F8.2,1X,'SOURCE HALF THICKNESS = ',1X,F8.2,/,
     210X,'SLAB HALF THICKNESS = ',1X,F8.2)
  223 FORMAT (/,10X,'FIRST COLLISION SOURCE CENTERED ABOUT X = 0.0',
     1/,T10,'SLAB HALF THICKNESS = ',2X,F8.2,/,
     210X,'DIFFUSION LENGTH = ',2X,F8.2)
 2231 FORMAT (/,10X,'FIRST COLLISION SOURCE CENTERED ABOUT X =',2X,
     1F8.2,2X,/,10X,'SLAB HALF THICKNESS=',2X,F8.2,/,
     210X,'DIFFUSION LENGTH = ',2X,F8.2)
  224 FORMAT (/,10X,'INPUT DATA IS PUNCHED CARDS')
```

```
225 FORMAT (/,10X,'INPUT DATA FROM MAGNETIC TAPE')
226 FORMAT (/,10X,'INPUT DATA FROM DISK')
227 FORMAT (/,10X,'OUTPUT WILL BE LISTED ON THE PRINTER')
228 FORMAT (/,10X,'OUTPUT PLOTS WILL BE PLOTTED ON THE PRINTER')
229 FORMAT (/,10X,'GRAPHICAL OUTPUT WILL BE PLOTTED USING THE',
   11X,'CAL COMP PLOTTER')
230 FORMAT (/,10X,'MODES OF DATA TO PROCESS =',2X,I5)
320 FORMAT (/,10X,'L(20)=1',T25,'SUBROUTINE-FOUT4 WILL BE',
   11X,'EXECUTED',/,T25,'UNCORRECTED SPECTRAL PLOT OF MOD-5 OUTPUT',
   22X,'--> F(U,T) VS U',/,T35,I5,3X,'HARMONIC MODES OF DATA')
321 FORMAT (/,10X,'L(17)=1',T25,'SUBROUTINE-FOUT5 WILL BE EXECUTED',
   1/,T25,'SPECTRAL PLOT OF MOD-5 OUTPUT ADJUSTED BY THE FOURIER',/,
   2T25,'EXPANSION COEFFICIENT AT THE POINT X = 0.0',
   35X,'FOR EACH OF',1X,I3,1X,'MODES',/,
   4T25,'AND A SUM OF ALL MODES WILL BE PLOTTED')
322 FORMAT (/,10X,'L(18)=1',T25,'SUBROUTINE-FOUT1 WILL BE EXECUTED',
   1/,T25,'CALCULATION OF THE FUNCTION-- F(U,T,X)=F(U,T)*PHI(X)',
   2/,T25,'FOR',1X,I3,1X,'DATA POINTS',5X,'WITH OUTPUT VIA DEVICE',
   31X,I3,/,T25,'LIMIT OF',2X,I3,2X,'PLOTS FOR GRAPHICAL OUTPUT',/,
   4T25,'DATA POINT POSITIONS ARE',/,10(10X,10(F6.2,4X),/))
323 FORMAT (/,10X,'L(14)=1',T25,'SUBROUTINE-FOUT2 WILL BE EXECUTED',
   1/,T25,'CALCULATION OF THE MEAN ENERGY AND FLUX-DENSITY PROFILE',
   2/,T25,'EMEAN(X,T) AND F(X,T) FOR EACH TIME STEP',/,
   3T25,'A TOTAL OF',2X,I3,1X,'PLOTS WITH BOTH FUNCTIONS WILL BE',
   41X,'DONE')
324 FORMAT (/,10X,'L(19)=1',T25,'SUBROUTINE-FOUT3 WILL BE EXECUTED',
   1/,T25,'CALCULATION OF A DETECTOR RESPONSE AT POSITION',1X,
   2F6.2,2X,'WITH',/,T25,'A HALF THICKNESS OF',2X,F5.2,/,
   3T25,'DETECTOR WILL RESPOND TO NEUTRON ENERGY-EMAX=',1X,1PE10.3,
   42X,'EMIN =',1X,1PE10.3,' EV',/,T25,'OUTPUT VIA DEVICE =',1X,
   5I5)
2250 FORMAT (/,10X,'RAMP FUNCTION-EXTERIOR SOURCE')
3211 FORMAT (/,10X,'L( 4)=1',T25,'A SPECTRAL PLOT WILL BE DONE FOR',
   12X,I5,2X,'POINTS',/,T20,'POINTS ARE:',/,10X,5(2X,F8.5,2X))
3231 FORMAT (/,10X,'L( 5)=1',T25,'NO PRINTED LISTING OF MEAN ENERGY',
   11X,'OR FLUX PROFILE WILL BE DONE ')
3232 FORMAT (/,10X,'L( 7)=1',T25,'OVER RIDE OF OUTPUT SEQUENCE',
   11X,'ONLY FLUX PROFILE WILL BE PLOTTED')
3233 FORMAT (/,10X,'L( 8)=1',T25,'OVER RIDE OF OUTPUT SEQUENCE',
   11X,'ONLY MEAN ENERGY RATIO WILL BE PLOTTED')
     RETURN
     END
```

```
      SUBROUTINE ESTIM1
C        REVISED  2  JUNE  1971
C
C     INITIAL ESTIMATING PROGRAM TO PROVIDE GUIDANCE ON TOTAL
C     ESTIMATED EXECUTION TIME WHILE ERROR TESTING THE VERSION
C     OF MIL SIX INITIAL PROGRAM CHECKOUT TO INFORM THE USER OF THE
C     ANTICIPATED PROGRAM EXECUTION PARAMETERS
C     EXECUTE THIS ROUTINE IMMEDIATELY AFTER READ10 OR READ 11
C     TO BEST ESTIMATE TOTAL EXECUTION TIME AFTER WE HAVE BETTER
C     KNOWLEDGE OF THE TOTAL PROGRAM OPERATING PARAMETERS
C
      COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
      COMMON/COM7/ SO1,SO2,SO3,SO4,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
      COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
      COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
     1NSTAT,MODE
      COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
      COMMON/COM6/ NTOP(6,20),NM(6,20),IKNT(6)
      REAL ST(15)/15*0.0/
      WRITE (6,76)
      WRITE (6,77)
C     DEFINE TOTAL TIME AS A ZERO VALUE PRIOR TO CALCULATION
      SUMT = 0.0
C     TIME FOR MAIN AND INCON1
      ST(1) = 0.75
C     EXECUTION TIME FROM READ PROGRAMS
      IF (( IDATA.EQ.1).AND.(NOUT.EQ.1)) ST(2)=MODES*1.25 + 0.50
      IF ((IDATA.EQ.1).AND.(NOUT.EQ.3)) ST(2)=MODES*1.0 + 0.50
      IF ((IDATA.EQ.2).AND.(NOUT.EQ.1)) ST(3)=MODES*1.25 + 0.50
      IF ((IDATA.EQ.2).AND.(NOUT.EQ.3)) ST(3)=MODES*1.0 + 0.40
C     FOUT1 TIME
      IF ((L(18).EQ.1).AND.(NOUT.EQ.1)) ST(4)=MODES*IBX*12.50
      IF ((L(18).EQ.1).AND.(NOUT.EQ.3)) ST(4)=MODES*IBX*2.50
C     FOUT2 TIME ESTIMATE
      IF (L(14).EQ.1) ST(5)=10.0*MODES
C     FOUT3 TIME ESTIMATE
      IF ((L(19).EQ.1).AND.(NOUT.EQ.1)) ST(6)=MODES*20.0
      IF ((L(19).EQ.1).AND.(NOUT.EQ.3)) ST(6)=MODES*5.00
C     ESTIMATED FOUT4 TIME FOR ECECUTION
      IF (L(20).EQ.1) ST(7)=MODES*3.0
C     FOUT5 TIME ESTIMATE
      IF (L(17).EQ.1) ST(8)=MODES*4.0
C     ESTIMATED TIME - ESTIM1
      ST(9)=0.25
C     ESTIMATED TIME - MODEL1
      ST(10)=0.25*MODES
```

```
C
C     SUM ALL TIME STEP ESTIMATES
      DO 63 IX2=1,10
   63 SUMT=SUMT+ST(IX2)
C     NOW WRITE OUT TOTAL TIME ESTIMATES
      WRITE (6,78) (ST(IA),IA=1,10),SUMT
   76 FORMAT ('1',//,30X,'ESTIMATED PROGRAM EXECUTION TIME')
   77 FORMAT (//,10X,'SUBROUTINE',10X,'ESTIMATED TIME',/,
     130X,'SECONDS')
   78 FORMAT (//,10X,'MAIN + INCON1',T30,F8.3,/,10X,'READ10',T30,F8.3,/,
     110X,'READ11',T30,F8.3,/,10X,'FOUT1',T30,F8.3,/,
     210X,'FOUT2',T30,F8.3,/,10X,'FOUT3',T30,F8.3,/,
     310X,'FOUT4',T30,F8.3,/,10X,'FOUT5',T30,F8.3,/,
     510X,'ESTIM1',T30,F8.3,/,10X,'MODEL1',T30,F8.3,///,
     610X,'TOTAL TIME = ',T30,F8.5)
      RETURN
      END


      FUNCTION PHI1(IMODE,IS,XPNT)
C        REVISED  10  JUNE  1971
C
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
      COMMON/COM7/ S01,S02,S03,S04,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
C
C     DEFINITIONS:
C        MODE  = NUMBER OF THE HARMONIC MODE
C        IS    =  TYPE SOURCE GERMOETRY
C
C     NOTE THAT ALL NON-HARMONIC DEPENDENT FACTORS ARE PRECALCULATED
C        IN SUBROUTINE - MODEL1  AND STORED IN COMMON BLOCK COM7
      X=XPNT
      GO TO (1,2,3,4,5),IS
    1 CONTINUE
C     IS=1
      MMOD=2*IMODE-1
      AKN1=MMOD*AK11
      PHI1= (S01/SOH)*SIN(AKN1*SA)*COS(AKN1*X)/AKN1
      GO TO 10
    2 CONTINUE
C     IS=2
      AKN2=(2*IMODE-1)*AK12
      PHI1=(AN2X)*SIN(AKN2*SA)*COS(AKN2*(X-S1))
      GO TO 10
    3 CONTINUE
C     IS=3
      M66M=(-1)**(IMODE+1)
```

```
      AKN3=AK13*IMODE
      AN3=AN3C*(AKN3*M66M*AN3B-AN3A)/(1.0+(AKN3*DIFLEN)**2)
      PHI1=AN3*COS(AKN3*X)
      GO TO 10
    4 CONTINUE
C     IS=4
      M44M=2*IMODE-1
      M55M=(-1)**(IMODE+1)
      AKN4=AK14*M44M
      AK5=AKN4*S1
      AK6=AKN4*DIFLEN
      AK7=1.0+(AKN4*DIFLEN)**2
      SIX5=SIN(AK5)
      CIS6=COS(AK5)
      AN4=(ANA4/AK7)*(CIS6-AK6*SIX5-(M55M*AK6*ANB4))
      CN4=(ANA4/AK7)*(SIX5-AK6*CIS6+M55M*ANB4)
      XPT=AKN4*X
      PHI1=AN4*COS(XPT)+CN4*SIN(XPT)
      GO TO 10
    5 CONTINUE
C     IS = 5
      PHI1=AK5X*SIN(IMODE*AK15*(XPNT+SOH))/(1.0*IMODE)
   10 CONTINUE
      RETURN
      END


      SUBROUTINE MODEL1
C        REVISED  12 JUNE  1971
C
C     THIS ROUTINE WILL:
C     1  CALCULATE THE CONSTANT ( NON-HARMONIC ) FACTORS IN THE
C        FOURIER POSITION EXPRESSIONS FOR THE INDICATED SOURCE
C        GEOMETRY( IS ) AS STATED IN THE PROBLEM
C     2  CALCULATE THE SUM OF THE INTERGRATED FOURIER EXPRESSION
C        TO DETERMINE THE TRUNCATION ERROR FOR THE REMAINING TERMS
C        OF THE FOURIER EXPANSION
C     3  ADJUST THE VALUE OF THE SOURCE STRENGTH ( SZERO ) TO OBTAIN
C        AN APPROXIMATION TO WITHIN + OR - 5% OF THE ORIGIONAL SOURCE
C        A LIMIT OF 10 INTERATIONS WILL BE DONE.
C        A DEFAULT OF 3 INTERATIONS WILL BE DONE IF THIS IS NOT
C        SPECIFIED BY THE VARIABLE NANS ( 4 ) IN THE INPUT DATA
C     4  IF LOGICAL L ( 9 ) = 1 A SIMPLE PLOT OF THE TOTAL SUM VS
C        POSITION ( X ) WILL BE DONE ON THE PRINTER
C        L( 9)      EXTERNAL CONTROL -PERFORM SIMPLE PLOT OF SOURCE
C                   FUNCTION ON THE PRINTER IN ROUTINE MODEL1
C        NANS( 4 ) - EXTERNAL CONTROL - LIMIT TO ALLOWABLE NUMBER
C                      OF INTERATIONS IN ROUTINE MODEL1
```

```
      COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
      COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
      COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
      COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
     1NSTAT,MODE
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
      COMMON/COM7/ SO1,SO2,SO3,SO4,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
      DIMENSION FX6(100)
      EQUIVALENCE (DUMMY,FX6)
C
C     DEFINE ALL TERMS FOR THE FOURIER EXPANSION COEFFICIENTS
C     AS INITIAL CONDITIONS TO REDUCE COMPUTATION IN THE FUNCTION
C     SUBPROGRAM PHI1 FOR ALL FOUR ( 5 ) SOURCE TYPE GEOMETRIES
C
      CALL CANCEL(2)
      IF(NANS(4).LE.0) NANS(4)=3
      BUCKLE=BUCK(1)
      WRITE (6,96)
      IF (NANS(4).GT.10) NANS(4)=10
C     PROTECTION STEP TO RETAIN INITIAL VALUE OF SZERO
      SPZERO=SZERO
      SP1P1= 1.10*SZERO
      SPZP9=0.90*SZERO
      NFLAG=0
      PI= 3.14159
C     START OF INTERATION LOOP
 1001 CONTINUE
      NFLAG=NFLAG+1
      WRITE (6,97) NFLAG,IS
C     BRANCHING RETURN OF INTERATIVE PROCEDURE
C     THE ROUTINE WILL DROP OUT OF THE INTERATION SERIES IF ONE OF
C     THE FOLLOWING CONDITIONS IS MEET:
C     1  CORRECTION FACTOR ( CORF ) IS BETWEEN 1.05 AND 0.95
C     2  SUM OF ADJUSTED TRUNCATED SERIES IS BETWEEN 1.10 AND 0.90 OF
C        PARAMETER - SZERO
C     3  TOTAL NUMBER OF INTERATION PASSES ( NFLAG ) EQUALS CONTROL
C        VALUE INPUT -- NANS ( 4 )
      GO TO (201,202,203,204,205),IS
  201 CONTINUE
C     IS=1
      SO1 = SZERO/2.0*SA
      AK11 = SQRT(BUCKLE)
      GO TO 215
  202 CONTINUE
C     IS=2
      SO2=SZERO/2.0*SA
      DELTA2=S1 + PI/(2.0*SQRT(BUCKLE))
```

```
      AK12= PI/(2.0*DELTA2)
      AK12=AK12*0.5
      AN2X= 4.0*SO2/PI
      GO TO 215
  203 CONTINUE
C     IS=3
      AK13= SQRT(BUCKLE)
      AK13=AK13*0.5
      AN3A= 1.0/DIFLEN
      AN3B=EXP(-1.0*SOH/DIFLEN)
      SO3= SZERO/(2.0*DIFLEN*(1.0 -AN3B))
      AN3C = (SO3/SOH)*((DIFLEN)**2)
      GO TO 215
  204 CONTINUE
C     IS=4
      SO4=SZERO/(2.0*DIFLEN)*(1.0-EXP(-1.0*(SOH+S1)/DIFLEN))
      DELTA4= S1 + PI/(2.0*SQRT(BUCKLE))
      AK14= PI/(2.0*DELTA4)
      AK14=AK14*0.25
      ANC4= EXP(-1.0*SOH/DIFLEN)
      ANB4= SINH(S1/DIFLEN)
      ANA4= 2.0*SO4*DIFLEN
      ANB4= ANB4*ANC4
      GO TO 215
C     IS=5
  205 CONTINUE
      AK15=SQRT(BUCKLE)
      AK5X=2.0*SZERO/(SOH*PI)
  215 CONTINUE
C     START APPROXIMATION ANALYSIS TO DETERMINE THE ERROR IN
C     FOURIER APPROXIMATION FOR THE TOTAL MODES OF HARMONIC DATA TO
C     CONSIDER FOR THIS PROBLEM
      GO TO (301,302,303,304,305),IS
  301 CONTINUE
C     IS=1 CASE
      SUM1 = 0.0
      DO 95 MODE=1,MODES
      AKN1=(2*MODE-1)*AK11
      SUM1 = SUM1 + SIN(AKN1*SOH)*SIN(AKN1*SA)/((AKN1)**2)
   95 CONTINUE
      SUM1 = SUM1*2.0*SO1/SOH
      ER1= SZERO-SUM1
      COR1= ER1/SUM1
      CORF=1.0+COR1
      WRITE (6,98) SZERO,IS,MODES,SUM1,ER1,COR1,CORF
      IF (( SUM1.LE.SP1P1).AND.(SUM1.GE.SPZP9)) GO TO 1015
      GO TO 315
  302 CONTINUE
```

```
C     IS=2 CASE
      SUM2 = 0.0
      DEL1 = SOH-S1
      DO 295 MODE=1,MODES
      AKN2= AK12*(2*MODE-1)
      BX2 = 2.0*SIN(AKN2*SOH)*COS(AKN2*S1)
      BX3 = SIN(AKN2*SA)/((2*MODE-1)**2)
      BX4 = SIN(AKN2*DEL1)
      BXSUM=BX3*(BX2-BX4)
      SUM2=SUM2 + BXSUM
  295 CONTINUE
      SUM2 = SUM2*SO2/(SA*(PI**2))
      ER2= SZERO-SUM2
      COR2= ER2/SUM2
      CORF=1.0+COR2
      WRITE (6,98) SZERO,IS,MODES,SUM2,ER2,COR2,CORF
      IF (( SUM2.LE.SP1P1).AND.(SUM2.GE.SPZP9)) GO TO 1015
      GO TO 315
  303 CONTINUE
C     IS= 3 CASE
      ABX3 = 4.0*SO3*((DIFLEN)**2)/(SOH*AK13)
      SUM3 = 0.0
      DO 395 MODE=1,MODES
      M35 = (-1)**(MODE+1)
      M36= 2*MODE - 1
      AKN3 = AK13*M36
      BX35 = M35*(AKN3*EXP(-1.0*SOH/DIFLEN)*M35-1.0/DIFLEN)
      BX36 = M36*(1.0+((AKN3*DIFLEN)**2))
      BX37 = BX35/BX36
      SUM3 = SUM3 + BX37
  395 CONTINUE
      SUM3 = SUM3*ABX3
      ERR3 = SZERO-SUM3
      COR3= ERR3/SUM3
      CORF=1.0+COR3
      WRITE (6,98) SZERO,IS,MODES,SUM3,ERR3,COR3,CORF
      IF (( SUM3.LE.SP1P1).AND.(SUM3.GE.SPZP9)) GO TO 1015
      GO TO 315
  304 CONTINUE
C     IS=4 CASE
      SUM4 = 0.0
      DO 495 MODE=1,MODES
      AKN4=AK14*(2*MODE-1)
      AX5=AKN4*S1
      AX6=AKN4*DIFLEN
      AX7=EXP(-1.0*SOH/DIFLEN)*SINH(S1/DIFLEN)
      BXS5=COS(AX5)-AX6*SIN(AX5)+((-1)**MODE)*AX6*AX7
      BXS6=SIN(AKN4*SOH)/(1.0+((AX6)**2))
```

```
      BX5=BXS5*BXS6
      SUM4=SUM4+BX5
  495 CONTINUE
      SUM4=SUM4*2.0*SO4*DIFLEN
      ER4=SZERO-SUM4
      COR4=ER4/SUM4
      CORF=1.0+COR4
      WRITE (6,98) SZERO,IS,MODES,SUM4,ER4,COR4,CORF
      IF (( SUM4.LE.SP1P1).AND.(SUM4.GE.SPZP9)) GO TO 1015
      GO TO 315
C     IS=5
  305 CONTINUE
      SUM5=0.0
      AX51=4.0*SZERO/(PI**2)
      DO 595 MODE=1,MODES
      M51=(-1)**(MODE+1)
      M52= 1+M51
      M53=MODE**2
  595 SUM5=SUM5+(1.0*M51*M52)/(1.0*M53)
      SUM5=SUM5*AX51
      ER5=SPZERO-SUM5
      COR5=ER5/SUM5
      CORF=1.0+COR5
      WRITE(6,98) SZERO,IS,MODES,SUM5,ER5,COR5,CORF
      IF((SUM5.LE.SP1P1).AND.(SUM5.GE.SPZP9)) GO TO 1015
  315 CONTINUE
      IF (( CORF.LE.1.05).AND.(CORF.GE..95)) GO TO 1015
      IF (NANS(4).EQ.NFLAG) GO TO 1015
      SZERO=SZERO*CORF
      GO TO 1001
 1015 CONTINUE
C
C     END OF ERROR ANALYSIS ROUTINE
C     A PLOT OF THE SUM OF ALL FOURIER POSITION DEPENDENT TERMS
C     WILL BE PLOTTED ON THE PRINTER IF THE LOGICAL CONTROL VARIABLE
C     L ( 9 ) = 1  TO GIVE A SIMPLE SOURCE REPRESENTATION
      IF ( L(9).EQ.0) GO TO 25
C     PROTECTION STEP TO ZERO ALL STORAGE PRIOR TO PLOTTING SUM
      DO 36 IK5=1,100
   36 FX6(IK5)=0.0
      NPNTS=100
      DELX=SOH/49.0
      TXBX(1)=-1.0*SOH
      TXBX(50)=0.0
      TXBX(100)=+SOH
      DO 4 IKX=2,49
      TXBX(IKX)=TXBX(1) + DELX*(IKX-1)
    4 CONTINUE
```

```
      DO 5 IKX=51,99
      TXBX(IKX)= DELX*(IKX-50)
    5 CONTINUE
      DO 6 NPT=1,100
      DO 6 IMODE=1,MODES
    6 FX6(NPT)=FX6(NPT)+PHI1(IMODE,IS,TXBX(NPT))
      MODC=0
      WRITE (6,1006) IS,IMODE,MODES
      CALL PLOTP(TXBX,FX6,NPNTS,MODC)
   25 CONTINUE
      SZERO=SPZERO
 5001 CONTINUE
   96 FORMAT ('1',///,T30,'SOURCE MODEL CORRECTIONS FOR FINITE SERIES')
   97 FORMAT (/,T20,'INTERATION NUMBER',2X,I3,2X,'FOR SOURCE TYPE',I5)
   98 FORMAT (///,T30,'ESTIMATED ERROR IN SOURCE MODEL',///
     1T30,'SOURCE STRENGTH',T65,1PE10.3,/,T30,'SOURCE TYPE',
     22X,'GEOMETRY',T65,I5,/, T30,'TOTAL MODES-FOURIER EXPANSION',
     3T65,I3,/,T30,'SUM OF FOURIER MODES',T65,1PE10.3,//,T30,
     4'ESTIMATED ERROR',T65,1PE10.3,/,T30,'FRACTIONAL ERROR',
     5T65,1PE10.3,/,T30,'SOURCE NORMALIZATION FACTOR',T65,1PE10.3)
 1006 FORMAT (//,20X,'SOURCE TYPE =',T40,I5,/,
     119X,'SUM OF FIRST',2X,I3,2X,'TERMS OF',2X,I3,2X,'MODES')
      RETURN
      END


      SUBROUTINE READ10(LASTN)
C        REVISED  2  JUNE  1971
C
C     INPUT ALL BASIC STATE PARAMETERS FROM MOD-5 PROBLEM OUTPUT
C     ECHO PRINT OF STATE STRUCTURE
C
C        READ10 PROCESSES ALL PUNCH CARD OUTPUT FROM MOD-5
C        BY READING A BLOCK OF SIX DATA SETS FUNCTION F(U,T)
C        FOR UP TO SIX MODES OF BUCKLING AND TWENTY TIME STEPS
C        OF DATA PER BUCKLING MODE
C        FOR THE INPUT DATA -- A PRINTED OUTPUT IS GIVEN OF THE
C        MOD 5 STATE STRUCTURE NU(IA) AND E(IA) FOR
C        ALL ENERGY/LETHERGY STATES OF THE  SYSTEM
C        THIS OUTPUT IS LISTED ONCE AFTER THE FIRST SET OF INPUT DATA
C        A PRINT OUT LISTING TIME STEP DATA  I.E. NT AND TIM(NT)
C        IS GIVEN AT THE END OF EACH DATA INPUT SEQUENCE
C        AFTER ALL DATA FOR THE PARTICULAR MODE IS DONE
C        ROUTINE DOES A TEST COUNT TO DETERMINE WHICH
C        DATA SETS WILL BE PROCESSSED BY CHECKING THOSE PROGRAMS
C        WHICH GAVE MOD 5 OUTPUT AT THE SAME VALUES OF NT
C        A LISTING IS GIVEN AT THE END OF THE READ SEQUENCE
C        OF THE NUMBER OF TIME STEPS THAT WILL BE RETAINED.
```

```
C
C         IF THIS ROUTINE IS REQUESTED TO PROCESS MORE THAN
C         TWENTY GROUPS OF TIME STEP DATA FOR ANY BUCKLING MODE
C         AN AUTO DEFAULT OPTION IS INCLUDED TO REJECT ALL EXCESS
C         AND ONLY USE THE FIRST 20 VALUES OF THE TIME DATA
C         ALL EXCESS VALUES ARE READ INTO A DUMMY ADDRESS AND IGNORED
C
      COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
      COMMON/COM7/ S01,S02,S03,S04,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
      COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
      COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
     1NSTAT,MODE
      COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
      COMMON/COM6/ NTOP(6,20),NM(6,20),IKNT(6)
      DIMENSION MTITLE(54)
C
C     READ10 ERROR DEFAULT OPTIONS
      LASTN = 0
      DO 15 ISET=1,MODES
C         THIS CALL SEQUENCE TO READ10  WILL RESULT IN THE
C         THREE TITLE CARDS AT THE START OF A DATA SET TO BE READ
      READ (5,43,ERR=20) N22
      WRITE (6,53)
      WRITE ( 6,57)  N22
      READ (5,44) (MTITLE(IA),IA=1,54)
      READ(5,45) N,NVIR,NI,NF,BUCKLE
C     PRINT OUT OF INPUT DATA - PROBLEM FROM MOD 5
      WRITE (6,50) (MTITLE(I),I=1,54)
      WRITE (6,51) N,NVIR,NI,NF,BUCKLE
C     ADD TO READ10    CHANGE LATER TO ADJUST TO PROCESSING OF MORE
C     THAN SIX MODES OF DATA
      BUCK(ISET) = BUCKLE
      IF (ISET.GT.1) GO TO 54
      READ(5,46)(U(IA),IA=1,NVIR)
      READ(5,46)(E(IA),IA=1,NVIR)
      WRITE (6,55)
      NX2 = NVIR/2
      NX3 = NX2 + 1
      DO 42 IB=1,NX3
      IBP=IB+NX2+1
      WRITE (6,49) IB,U(IB),E(IB),IBP,U(IBP),E(IBP)
      IF (IBP.GE.NVIR) GO TO 54
   42 CONTINUE
   54 CONTINUE
   62 CONTINUE
C     SET PROGRAM TO READ A MAXIMUM OF 20 TIME INTERVALS
```

```
C     THEN DEFAULT BY PRINTING OUT ALL REMAINING DATA THAT
C     CANNOT BE PROCESSED INTO MIL SIX
C     MOD FIVE  PRODUCES THE FUNCTIONS  F(U,T)
C     DUMMY PARAMETER NRUN  DEFINES INTERMEDIATE OR LAST DATA SET
C     IF NRUN = 1  THIS IS INTERMEDIATE DATA SET
C     IF NRUN = 2  THIS IS LAST DATA SET
C
      IA=ISET
      NTEST = NCALL
      NCALL = 0
      IKNT(ISET) = 0
      WRITE (6,23)
   21 NCALL = NCALL + 1
      IF (NCALL.EQ.21) GO TO 8
      READ (5,1,END=13,ERR=16) NRUN,TIM(NCALL),NT(NCALL),
     1NTOP(IA,NCALL),NM(IA,NCALL)
C     MOD-5 ALWAYS PUNCHES CARDS FROM THE DENSITY VECTOR IN THE
C     FORM  (POP(IA,1),IA=1,NM)  IN A FORMAT 1PE10.3 FOR 4 FIGURE DATA
      MIL1=NM(IA,NCALL)
      IF (NRUN.EQ.1) IKNT(ISET) = IKNT(ISET) + 1
C
C     TEST SERIES TO DETERMINE THE TOTAL DATA SETS WITH NRUN = 1
C     THIS SHOULD REDUCE THE PROCESSING OF USELESS DATA LATER IN THE
C     PROGRAM CALCULATIONS
C     OUTPUT NEWS NOTE TO WRITE AT  END OF THE CARD READING ROUTINE
      READ (5,2,END=15,ERR=16) (FUT(IA,IB,NCALL),IB=1,MIL1)
      WRITE (6,5) NRUN,TIM(NCALL),NT(NCALL),NTOP(IA,NCALL),NM(IA,NCALL)
      IF (NRUN.EQ.1) GO TO 21
    3 CONTINUE
      IF (NRUN.EQ.2) WRITE(6,7)
C
      GO TO 15
    8 CONTINUE
C
C     THIS FIXES DATA SET 20 AS THE MAXIMUM AMOUNT FOR THE PROGRAM
C
      MTEST = 0
    9 WRITE (6,10)
   11 READ (5,1,END=13,ERR=16) NNRUN,TIMN,NTTN,NTOPN,NMM
      WRITE (6,5) NNRUN,TIMN,NTTN,NTOPN,NMM
      IF (NNRUN.EQ.1) MTEST = MTEST + 1
      READ (5,2) (DUMMY(IA),IA=1,NVIR)
   12 IF (NNRUN.EQ.1) GO TO 11
   13 WRITE (6,7)
      IF (NTEST.GE.NCALL)  NCALL=NTEST
   15 CONTINUE
      GO TO 19
   20 CONTINUE
```

```
C
   16 LASTN=1
C
   19 CONTINUE
      WRITE (6,91) MODES
      DO 90 IX1 = 1,MODES
      WRITE (6,92) IX1,IKNT(IX1)
   90 CONTINUE
C
    1 FORMAT (I5,E15.8,3I5)
    2 FORMAT(8E10.3)
    5 FORMAT (10X,I5,5X,1PE10.3,5X,3I5)
    7 FORMAT (///,10X,'END OF INPUT DATA FILE - F(U,T)')
   10 FORMAT(///,10X,'NCALL EXCEEDS  20 - MAXIMUM SPACE IN PROGRAM')
   23 FORMAT (//,10X,'F(U,T)-TIME STEP DATA',//,11X,'NRUN',
     19X,'TIM(NT)',6X,'NT',2X,'NTOP',3X,'NM')
   43 FORMAT (I5)
   44 FORMAT(18A4)
   45 FORMAT (4I5,E15.8)
   46 FORMAT(8(1PE10.3))
   49 FORMAT (10X,2(I5,5X,2(1PE10.3,8X),6X))
   50 FORMAT (10X,18A4)
   51 FORMAT (10X,'NUMBER OF REAL STATES =',3X,I5,/,
     110X,'NUMBER OF VIRTUAL STATES =',I5,/,
     210X,'NUMBER OF ISOTOPES =',9X,I5,/,
     310X,'NUMBER OF FISSILE ISOTOPES =',2X,I5,/,
     410X,'BUCKLING =',15X,F10.5)
   53 FORMAT ('1',10X,'INPUT DATA FROM MOD 5')
   55 FORMAT (//,8X,(2('STATE',9X,'U-LETHARGY',6X,'ENERGY--EV',5X)))
   56 FORMAT (/,10X,I5)
   57 FORMAT (10X,I5)
   91 FORMAT (///,10X,I5,2X,'TOTAL USEFUL DATA SETS FROM MOD-5',/,
     115X,'DATA SET',5X,'SETS')
   92 FORMAT (17X,I3,7X,I3)
      RETURN
      END


      SUBROUTINE FOUT1(XPNT)
C        REVISED  2  JUNE  1971
C
C        SUBROUTINE FOUT1
C     CALCULATES THE POSITION FUNCTION F(U,T,X) FOR A SERIES OF SIX
C     BUCKLING SETS AT A TIME.
C     IF ONLY SIX BUCKLING MODES ARE CONSIDERED,THIS ROUTINE CAN DO
C     THE TIME-ENERGY-POSITION FUNCTION FOR AS MANY POINTS AS DESIRED
C     OF THIS ONE DIMENSIONAL MODEL -- SLAB GEOMERTY
C
```

```
C     FUTURE CHANGES FOR THIS ROUTINE WILL BE ABLE TODEAL WITH
C     ALMOST ANY QUANTITY OF INPUT DATA BY THE EARLIER CALCULATION
C     OF A NEW FUNCTION CALLED  PHIX(NN,MM)
C     THIS FUNCTION WILL BE BASED ON A TOTAL NUMBER OF POINTS FOR
C     SOME MAXIMUM NUMBER OF BUCKLING MODES--MM
C     FUTURE PROCESSING WILL FOLLOW THE SEQUENCE
C     1  DEFINE TOTAL MODES OF DATA TO PROCESS
C     2  CALCULATE THE POSITION FUNCTION PHIX(NX,NMODES)
C     3  CALCULATE/UPDATE THE FUNCTION
C        F(NU,NX,NT) = F(NU,NT,IMODE)*PHIX(NX,IMODE) +F(U,T,X)-OLD
C     4  TERMINATE THE CALCULATE RUN WHEN ALL DATE HAS BEEN EXHAUSTED
C     5  PROVIDE OUTPUT VIA  THREE DEVICES
C                                          PRINTER-LIST RESULTS
C                                          PRINTER-PLOT RESULTS
C                                          PLOTTER-DRAW GRAPHS/CHARTS
C        MODIFICATION WILL REQUIRE THE ADDITION OF TWO ARRAYS
C        1     PHIX(NXPNT,NXMODES)
C        2     AFUTX(NXPNT,NU,NT)
C     CURRENT FUNCTION -- FUTX(NU,NT) DEALS WITH ONLY ONE POSITION
C     VALUE OF THE POSITION POINT-'X'   AT THIS TIME
C
      COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
      COMMON/COM7/ SO1,SO2,SO3,SO4,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
      COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
      COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
     1NSTAT,MODE
      COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
      COMMON/COM6/ NTOP(6,20),NM(6,20),IKNT(6)
C
      REAL INMM
      REAL LABEL(20)/'T1','T2','T3','T4','T5','T6','T7','T8','T9',
     1'T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'/
      REAL*8 TIMST(4)/'NT= 1-5','NT= 6-10','NT=11-15','NT=16-20'/
      REAL*8 APNT(6)/'POINT- 1','POINT- 2','POINT- 3','POINT- 4',
     1'POINT- 5','POINT- 6'/
      REAL*8 BMODES(20)/' 1-MODE',' 2-MODES',' 3-MODES',' 4-MODES',
     1' 5-MODES',' 6-MODES',' 7-MODES',' 8-MODES',' 9-MODES',
     2'10-MODES','11-MODES','12-MODES','13-MODES','14-MODES',
     3'15-MODES','16-MODES','17-MODES','18-MODES','19-MODES',
     4'20-MODES'/
      REAL*8 INAME(12)/' MILLS-B','OX-M    ','F(U,T,X)','-FLUX AT',
     1'X  VS U-','LETHRGY ','MOD-2 CA','LCULATIO','N       ',
     2'        ','        '/
      DIMENSION FFUTX(150)
      EQUIVALENCE (FFUTX,DUMMY)
C
```

```
C     CALCULATE THE FUNCTION F(U,T,X) = F(U,T)*PHI1(X) FOR ALL FUT
      N5= NSTAT
      X1 = XPNT
C
C     CHANGE TO FOUT1   CHANGE AFTER ROUTINE IS SET TO PROCESS MORE
C     THAN SIX INPUT DATA DECKS OR TAPE BY PROVIDING PROTECTED
C     STORAGE FOR THE FIRST MODE OF BUCKLING
C
      BUCK1 = BUCK(1)
      DO 2 IMOD=1,MODES
      NCALL1  =  IKNT(IMOD)
      DO 2 NCX=1,NCALL1
      MIL1 = NTOP(IMOD,NCX) + 1
      MIL2 = MIL1 + NM(IMOD,NCX)
      DO 2 M=MIL1,MIL2
      FUTX(M,NCX)=FUT(IMOD,M,NCX)*PHI1(IMOD,IS,X1)
     1 + FUTX(M,NCX)
    2 CONTINUE
C
C     OVER RIDE STEP TO BY PASS THE OUTPUT ROUTINES ON CALLS FROM
C     OTHER SUPROUTINE AND SUB PROGRAMS OF MOD 2
C     OVER RIDE PARAMETER IS LOGICAL -- L( 16)
C     IF  L(16) = 1   THE PRINT ROUTINE AND ALL OTHERS ARE SKIPPED
      IF (L(16).EQ.1)  GO TO 25
C
C     WRITE OUT ALL VALUES OF FUTX(M,NCX)
C     BEFORE CALLING THE CAL COMP PLOTTER
      NCALL = IKNT(1)
      WRITE (6,90) X,NVIR,NCALL
C
C     PLOT P SEQUENCE
      DO 93 II2=1,NCALL
      WRITE (6,94) II2,TIM(II2),NT(II2)
      WRITE (6,95) (FUTX(II3,II2),II3=1,NVIR)
   93 CONTINUE
      IF(NOUT.NE.3) GO TO 25
C     IF THE TEST VARIABLE NOUT NOT EQ 3 BY PASS REST OF SEQUENCE
C     DEFINE SCALE PARAMETERS FOR THE DRAW PLOT ROUTINE
C     PAGES TO BE PLOTTED VIA THE CAL COMP
C     TOTAL LIMITS ON GRAPH PLOTTING FOR THIS ROUTINE IS DEFINED
C     VIA THE INPUT CONTROL VARIABLE NANS (1) = (( THE TOTAL GRAPH
C     DEFINE A DEFAULT VALUE FOR NANS( 1 ) OF 1
      IF (NANS(1).LT.1) NANS(1) = 1
      EXSCAL = U(NVIR-1)/8.
      YSCALE = FUTX(2,1)/8.
      IXUP = C
      IYRIGH = C
      MODXAX = 0
```

```
      MODYAX = 0
      IWIDE = 9
      IHIGH = 10
      IGRID = 1
      KKNT = NCALL
      KTEST = NCALL/5 + 1
      KK1 = 0
      DO 8 IKT = 1,KTEST
C     MODIFICATION OF GRAPH TITLING TO INDICATE DATA SET PLOTTED
      INAME(10) = BMODES(MODES)
      NPXNT = NANS(2)
      INAME(11) = APNT(NPXNT)
      IF(IKT.LE.4) INAME(12) = TIMST(IKT)
      IF (IKT.GT.4) INAME(12) = TIMST(4)
      DO 5 IXM= 1,5
      KK1 = KK1 + 1
      MODCUR = 2
      IF (IXM.EQ.1) MODCUR = 1
      IF (IXM.EQ.5) MODCUR = 3
      IF (KK1.EQ.NCALL) MODCUR=3
      IF (KK1.GT.20) GO TO 3
      INMM = LABEL(KK1)
    3 IF(KK1.GT.20) INMM = LABEL(20)
C
C     CALL DRAW USING THE WORKING VECTOR FFUTX
C     USE OF THE INPUT CONTROL PARAMETER LIMITS GRAPH
C     TO SOME MULTIPLE OF FIVE DATA PLOTS IF POINT PLOTTING
C     PARAMETER USED IS NANS(1)
C     TEST QUESTION TO CHECK :  IF( NANS(1).EQ.1.AND.IKT.EQ.6)) GT TO10
C     INITIAL TEST QUESTION FORM:::::
C     SECOND FORM TO USE IN TI THE FUTURE:::
C           NTST2  =  NANS(1)*5 + 1
C        IF ( NTST2.EQ.IKT ) GO TO 10
C        PARAMETER  NTST2 DEFINES THE TOTAL NUMBER OF
C        TIMES  MOD5 THAT PLOT WILL BE CALLED
C
C     THREE DEFUALT OPTIONS TO DROP OUT OF THE CALL DRAW PROGRAM
C
      DO 4 ICX = 1,NVIR
    4 FFUTX(ICX) = FUTX(ICX,KK1)
      IF ((NANS(1).EQ.1).AND.(IKT.EQ.6)) GO TO 10
      IKT1 = IKT
      IF (NVIR.GE.30) IKT1 = 0
C
      CALL DRAW (NVIP,U,FFUTX,MODCUR,IKT1,INMM,INAME,EXSCAL,YSCALE,
     1IXUP,IYRIGH,MODXAX,MODYAX,IWIDE,IHIGH,IGRID,LAST)
      IF (LAST.EQ.2) GO TO 10
      IF (LAST.EQ.3) GO TO 10
```

```
      IF (KK1.EQ.NCALL) GO TO 10
C     ON REACHING MAXIMUM GRAPH OUTPUT-DROP OUT OF THE LOOP
      IF (NANS(1).GE.IKT) GO TO 10
    5 CONTINUE
    8 CONTINUE
   10 CONTINUE
   25 CONTINUE
   90 FORMAT ('1',10X,'CALCULATION OF THE POSITION FUNCTIONF(U,T,X)',
     1/,20X,'FOR X =',2X,F10.5,2X,'WITH',2X,I5,2X,'STATES AT',2X,
     2I5,2X,'TIMES')
   94 FORMAT (//,10X,'RUN NUMBER',2X,I5,2X,'TIME =',2X,E15.8,
     12X,'SECONDS',5X,'NT = ',2X,I5)
   95 FORMAT (41(/,10X,5(1PE10.3,5X)))
      RETURN
      END


      SUBROUTINE FOUT2
C        REVISED   16  JUNE 1971
C
C     CORRECTED ROUTINE TO CALCULATE THE MEAN ENERGY OF THE
C     NEUTRONS AT A POINTX(I)  AND THE SUM OVER ALL LETHARGY STATES
C     OF THE NEUTRON DISTRIBUTION AT THE SAME POINT AND PLOT BOTH
C     RESULTANT ON A SINGLE GRAPH FOR EACH TIME STEP VALUE
C     LETHARGY SUM
C        F(U,T,X,MODE) = F(U,T,MODE)*PHI(X,MODE)
C        F(U,T,X) = SUM OVER MODES--F(U,T,X,MODE)
C     OVER ALL STATES U(LETHERGY) TO GIVE THE RESULTANT
C     FUNCTION  --  F ( X,T )
C        F(X,T) = SUM OVER LETHARGY - F(U,T,X)
C     MEAN ENERGY CALCULATION
C        EMEAN(X,T) = SUM OVER LETHARGY-F(U,T,X)*E(U)
C     COMMON BLOCKS 3 AND 6  MUST BE REVISED TO HANDLE MORE THAN
C     SIX HARMONIC MODES OF THE FUNCTIONAL DATA INPUT
      DIMENSION VSIG(150)
      EQUIVALENCE (DUMMY(1),VSIG(1))
      REAL LABEL
      REAL*8 ENGY/'ENERGY'/,ENGX/'RATIO'/
      REAL*8 FNGY/'NEUTRON'/,FNGX/'DENSITY'/
      REAL*8 ARUNT(9)/'SYSTEM-1','SYSTEM-2','SYSTEM-3','SYSTEM-4',
     1'SYSTEM-5','SYSTEM-6','SYSTEM-7','SYSTEM-8','SYSTEM-9'/
      REAL*8 MTIT(12)/'FUNCTION',' F( X,T',' )     ','  ','TOM MILL',
     1'S BOX M','  ','  ','  ','  ','  ','  '/
      REAL*8 TIMX(20)/'NT - 1','NT - 2','NT - 3','NT - 4','NT - 5',
     1'NT - 6','NT - 7','NT - 8','NT - 9','NT -10','NT -11','NT -12',
     2'NT -13','NT -14','NT -15','NT -16','NT -17','NT -18',
     3'NT -19','NT -20'/
      REAL*8 BMODES(6)/'1-MODE','2-MODES','3-MODES','4-MODES',
```

```
     1'5-MODES','6-MODES'/
      DIMENSION EMEAN(100,20)
      COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
      COMMON/COM7/ S01,S02,S03,S04,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
      COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
      COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
     1NSTAT,MODE
      COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
      COMMON/COM6/ NTOP(6,20),NM(6,20),IKNT(6)
      CALL CANCEL(2)
C     PROGRAM OUTPUT CONTROL NANS( 3 ) = NN --- DEFINES THE TOTAL
C     NUMBER OF TIME STEP VALUES THAT THIS ROUTINE WILL PLOT
C     INITIAL DEFAULT OPTION WILL BE FOR ONE PLOT,IF THE USER FORGETS
C     TO DEFINE THIS PARAMETER IN THE INPUT ROUTINE
      IF (NANS(3).LE.0) KPLOT=1
      IF (NANS(3).GE.1) KPLOT=NANS(3)
C     CP1
      WRITE (6,1001)
C
C        DESCRIPTIONS OF THE POSITIONS ON THE X-AXIS FOR SOURCES
C        LOCATED OR CENTERED AT THE AXIS OF SYMMETRY
C     CALCULATE THE VELOCITY OF THE NEUTRONS FROM ENERGY DATA
C     VELOCITY = SQRT( 2.0*E(IPX)/MASS))
      DO 1057 IPX= 1,NVIR
      VSIG(IPX)=SQRT((1.918E12)*E(IPX))
 1057 CONTINUE
C     INPUT PARAMETER NANS( 2 ) IS USED TO PROVIDE  MODIFICATION OF
C     CAL COMP GRAPH LABELING  FOR UP TO NINE SYSTEMS
      LPNT=NANS(2)
      IF (( LPNT.LE.1).OR.(LPNT.GE.10)) LPNT=1
      MTIT(7)=ARUNT(LPNT)
      IF ((IS.EQ.2).OR.(IS.EQ.4)) GO TO 315
      IF(IS.EQ.5) GO TO 315
      NPNTS=100
      DELX=SOH/99.0
      DO 2 I=1,98
      IP=I+1
      TXBX(IP)=DELX*I
    2 CONTINUE
      TXBX(1)=0.0
      TXBX(100)=SOH
  315 CONTINUE
C     DESCRIPTION OF THE X-POSITIONS WITHIN THE LIMITS OF +/-SOH
C        FOR SOURCES LOCATED AT AN OFF-AXIS LOCATION
C        THIS WILL PROVIDE FOR ALL UNSYMMETRICAL POSITIONS IN THE SLAB
      IF ((IS.EQ.1).OR.(IS.EQ.3)) GO TO 316
```

```
      NPNTS=99
      DELX2=SOH/49.0
      TXBX(1)=(-1.)*SOH
      TXBX(99)=SOH
      DO 3 I2=2,98
      TXBX(I2)=TXBX(1)+DELX2*(I2-1)
    3 CONTINUE
      TXBX(50)=0.0
  316 CONTINUE
C     CP2
      WRITE (6,1002)
C     DEFINE INITIAL CONSTANTS FOR THE ROUTINE
      NONE = 1
      BK1=BUCK(1)
      NTMAX=IKNT(1)
      NMAX = NVIR-1
C     IMPROVEMENT TO INCREASE EXECUTION SPEED OF CALCULATION
      DO 321 NXPT=1,NPNTS
      DO 321 KMOD=1,MODES
      FUTX(NXPT,KMOD)=PHI1(KMOD,IS,TXBX(NXPT))
  321 CONTINUE
C     CP3
      WRITE (6,1003)
C     START OF CALCULATIONS
      DO 250 INT=1,NTMAX
      DO 245 IXPT=1,NPNTS
      AX=0.0
      BX=0.0
      CX5=0.0
      DO 240 IMODE=1,MODES
C        PROTECTION STEP WITHIN CALCULATION LOOP TO REDUCE UNNECESSARY
C        CALCULATIONS
      NP3=IKNT(IMODE)
      IF(INT.GT.NP3) GO TO 240
      MIL1=NTOP(IMODE,INT)
      MIL2=NM(IMODE,INT)
      MIL3= MIL1+MIL2
      DO 235 NU1=1,MIL2
      MIL4=MIL1+NU1
      IF (MIL4.GT.NMAX) GO TO 235
      BX1=FUT(IMODE,NU1,INT)*FUTX(IXPT,IMODE)
C
      CX5=CX5+BX1
      AX=AX+BX1*VSIG(MIL4)
      BX=BX+BX1*E(MIL4)
  235 CONTINUE
  240 CONTINUE
C        DEFAULT PROTECTION STEP FOR PARAMETER  CX5
```

```
C         THIS VARIABLE IS TO REPRESENT THE NEUTRON POPULATION DENSITY
C         FOR A PARTICULAR TIME STEP VALUE
C         IN GENERAL,IT SHOULD BE A VALUE BETWEEN 0.001 AND 1.000
C         FOLLOWING IS TO PROTECT AGAINST AN ACCIDENTAL ATTEMPT TO DEVIDE
C         BY ZERO  OR  SOME OTHER UNUSUAL VALUE THAT MIGHT ACCIDENTLY
C         COME UP.
C         TEST CHECK AND DEFAULT SET:
      IF ((CX5.LE.0.001).OR.(CX5.GE.1.001)) CX5=1.000
C
      EMEAN(IXPT,INT)=BX/CX5
      FXT(IXPT,INT)=AX/CX5
  245 CONTINUE
  250 CONTINUE
C     CP4
      WRITE (6,1004)
      DO 253 NT4=1,NTMAX
      IF ((IS.EQ.1).OR.(IS.EQ.3)) ZA=EMEAN(1,NT4)
      IF ((IS.EQ.2).OR.(IS.EQ.4).OR.(IS.EQ.5)) ZA=EMEAN(50,NT4)
C     DEFAULT TEST CHECK ON ZA VALUE
      WRITE (6,6001)  NT4,ZA
C
      IF (ZA.LE.0.0) GO TO 255
      DO 252 IX4=2,NPNTS
      FUTX(IX4,NT4) = EMEAN(IX4,NT4)/ZA
  252 CONTINUE
      FUTX(1,NT4) = 1.0
      GO TO 253
  255 WRITE (6,271) NT4,ZA
      DO 256 IX5=1,NPNTS
  256 FUTX(IX5,NT4) = .90
C
  253 CONTINUE
C     CP5
      WRITE (6,1005)
C     BY PASS THE CAL COMP ROUTINE PORTION IF NOUT NOT EQUAL 3
      IF ( NOUT.NE.3) GO TO 12
C         START PREPARATION OF DATA FOR GRAPH PLOTTING ROUTINES
C         TWO ITEMS OF DATA WILL BE PLOTTED FOR EACH TIME STEP
      LAST=0
      ITYPE=0
      XCAL=SOH/8.0
      IHIGH=10
      IWIDE=8
      IYRT=0
      IGRID=1
C     FXT(I,J) = FLUX PROFILE FOR X AND T
C     EMEAN(I,J) = MEAN ENERGY DATA VERSUS X AND T
C     FUTX(I,J) = RATIO MEAN ENERGY(X,T)/MEAN ENERGY(0.0,T)
```

```
C      L(K) =   1   EXTERNAL CONTROL -BYPASS THE PRINT SEQUENCE IN FOUT2
C      L(7) =   1   EXTERNAL CONTROL - PLOT FLUX PROFILE VS X ON CALCOMP
C      L(8) =   1   PLOT MEAN ENERGY RATIO VS X ON CAL COMP IN FOUT2
C      ADJUSTMENT OF TITLING FOR GRAPHS
       MTIT(10) = BMODES(MODES)
C      LOAD DUMMY WITH FXT(NX,NT) DATA
       DO 155 NXT1=1,NTMAX
       LABEL=LTAB(NXT1)
       MTIT(12)= TIMX(NXT1)
       IXUP1=0
       YSCAL1=FXT(1,NXT1)/4.0
       MODC1=1
       MODX1=0
       IXUP1=0
       IYRT=0
       MODY1=0
C      PROTECTION STEP FOR GRAPH PLOTTING
C      PLOTTING PROTECTION STEP TO MAINTAIN ONLY POSITIVE VALUES
       IF (L(7).EQ.0) GO TO 1175
       DO 162 NP5=1,NPNTS
       DUMMY(NP5)=FXT(NP5,NXT1)
       IF (DUMMY(NP5).LE.0.001) DUMMY(NP5)=0.0
  162 CONTINUE
C      CALL DRAW TO PLOT FIRST GRAPH FOR THIS TIME STEP
       MTIT(2)=FNGY
       MTIT(3)=FNGX
       CALL DRAW(NPNTS,TXBX,DUMMY,MODC1,ITYPE,LABEL,MTIT,XCAL,
      1YSCAL1,IXUP1,IYRT,MODX1,MODY1,IWIDE,IHIGH,IGRID,LAST)
       IF (LAST.EQ.1) GO TO 155
 1175 IF (L(8).EQ.0) GO TO 1176
       MODX2=2
       YSCAL2=0.50
       IXUP2=6
       MODC2=3
       MODY2=0
       DO 163 NP6=1,NPNTS
  163 DUMMY(NP6)=FUTX(NP6,NXT1)
C      CALL TO DRAW TO PLOT THE MEAN ENERGY RATIO
       MTIT(2)=ENGY
       MTIT(3)=ENGX
       CALL DRAW(NPNTS,TXBX,DUMMY,MODC2,ITYPE,LABEL,MTIT,XCAL,
      1YSCAL2,IXUP2,IYRT,MODX2,MODY2,IWIDE,IHIGH,IGRID,LAST)
C      CP6
C      INTERNAL PROTECTION STEP TO LIMIT TOTAL GRAPHICAL OUTPUT TO
C      THE CAL COMP PLOTTER VIA VARIABLE NANS ( 3 ) TO COMPARE
C      WITH THE INDEXING VARIABLE -- NXT1
C      KPLOTS IS A DUMMY VARIABLE DEFINED WITH A DEFAULT VALUE OF 1
 1176 CONTINUE
```

```
      IF ( NXT1.GE.KPLOTS ) GO TO 156
  155 CONTINUE
  156 CONTINUE
      WRITE (6,1006)
C     PLOTP OUTPUT ROUTINE USING THE PRINTER
   12 IF (NOUT.EQ.3) GO TO 16
      NCALL = IKNT(1)
      DO 15 J=2,NCALL
      DO 13 I=1,NPNTS
   13 DUMMY(I) = FXT(I,J)
      MODC = 0
      CALL PLOTP(TXBX,DUMMY,NPNTS,MODC)
      WRITE (6,14) J,NT(J),TIM(J)
   15 CONTINUE
C     PRINT-PLOT THE MEAN ENERGY VALUES OF THE PRINTER
      DO 615 J=2,NCALL
      DO 613 I=1,NPNTS
  613 DUMMY(I)= EMEAN(I,J)
      MODC=0
      CALL PLOTP(TXBX,DUMMY,NPNTS,MODC)
      WRITE (6,617) J,NT(J),TIM(J)
  615 CONTINUE
C     PRINT-PLOT THE MEAN ENERGY RATIO
      DO 715 J=2,NCALL
      DO 713 I=1,NPNTS
  713 DUMMY(I) = FUTX(I,J)
      MODC=0
      CALL PLOTP(TXBX,DUMMY,NPNTS,MODC)
      WRITE (6,717) J,NT(4),TIM(J)
  715 CONTINUE
C     PRINT OUT RESULTS ON THE PRINTER ONLY
C     PRINT OUT ALL VALUES OUT IN COLUMNS OF SIX TIME STEPS
C     L(5) =   1   EXTERNAL CONTROL -BYPASS THE PRINT SEQUENCE IN FOUT2
   16 IF(L(5).EQ.1) GO TO 23
C     OVER RIDE CONTROL TO SKIP THE PRINT SEQUENCE
   35 CONTINUE
      NCALL = IKNT(1)
      IF(NCALL.GT.6) NTT=6
      IF(NCALL.LE.6) NTT=NCALL
      WRITE (6,22) (NT(I),I=1,NTT),(TIM(I),I=1,NTT)
      WRITE (6,501)
      DO 17 I=1,NPNTS
   17 WRITE(6,21) TXBX(I),(FXT(I,J),J=1,NTT)
      WRITE (6,502)
      DO 67 I2=1,NPNTS
   67 WRITE (6,21) TXBX(I2),(EMEAN(I2,J2),J2=1,NTT)
      WRITE (6,503)
      DO 93 I2=1,NPNTS
```

```
93 WRITE (6,21) TXBX(I2),(FUTX(I2,J2),J2=1,NTT)
   IF(NCALL.LE.6) GO TO 23
   IF (NCALL.GT.12) NTT=12
   IF(NCALL.LE.12) NTT = NCALL
   WRITE (6,22) (NT(I),I=7,NTT),(TIM(J),J=7,NTT)
   WRITE (6,501)
   DO 18 I=1,NPNTS
18 WRITE (6,21) TXBX(I),(FXT(I,J),J=7,NTT)
   WRITE (6,502)
   DO 68 I2=1,NPNTS
68 WRITE (6,21) TXBX(I2),(EMEAN(I2,J2),J2=7,NTT)
   WRITE (6,503)
   DO 94 I2=1,NPNTS
94 WRITE (6,21) TXBX(I2),(FUTX(I2,J2),J2=7,NTT)
   IF (NCALL.LE.12) GO TO 23
   IF (NCALL.GT.18) NTT=18
   IF (NCALL.LE.18) NTT = NCALL
   WRITE (6,22) (NT(I),I=13,NTT),(TIM(J),J=13,NTT)
   WRITE (6,501)
   DO 19 I=1,NPNTS
19 WRITE (6,21) TXBX(I),(FXT(I,J),J=13,NTT)
   WRITE (6,502)
   DO 69 I2=1,NPNTS
69 WRITE(6,21) TXBX(I2),(EMEAN(I2,J2),J2=13,NTT)
   WRITE (6,503)
   DO 95 I2=1,NPNTS
95 WRITE (6,21) TXBX(I2),(FUTX(I2,J2),J2=13,NTT)
   IF (NCALL.LE.18) GO TO 23
   NXT = 0
   WRITE (6,22) NT(19),NT(20),NXT,NXT,NXT,NXT,TIM(19),TIM(20)
   WRITE (6,501)
   DO 20 I=1,NPNTS
20 WRITE(6,21) TXBX(I),FXT(I,19),FXT(I,20)
   WRITE (6,502)
   NTT=NCALL
   DO 70 I2=1,NPNTS
70 WRITE (6,21) TXBX(I2),(EMEAN(I2,J2),J2=19,NTT)
   WRITE (6,503)
   DO 96 I2=1,NPNTS
96 WRITE (6,21) TXBX(I2),(FUTX(I2,J2),J2=19,NTT)
23 CONTINUE
11 FORMAT (///,10X,I5,2X,'TOTAL POINT S PLOTTED USING DRAW',/,
  110X,I5,2X,'TOTAL CALLS MADE TO DRAW')
14 FORMAT (//,10X,'DATA PLOT NUMBER',2X,I5,/,10X,'TEST RUN NT =',
  12X,I5,/,10X,'TIME = ',2X,E11.4,2X,'SECONDS')
21 FORMAT (11X,F6.2,6X,6(1PE10.3,4X))
22 FCRMAT ('1',//,11X,'X = ',8X,6(4X,'NT = ',I3,1X),//,
  122X,6(1X,E10.3,3X),/,22X,6(5X,'SEC',6X))
```

```
  271 FORMAT (10X,'ERROR CHECK',2X,'NT = ',I4,2X,'EMEAN(1,NT) = ',
     12X,1PE10.3)
  501 FORMAT (//,10X,'FUNCTION-F( X,T )--SUM OVER LETHARGY')
  502 FORMAT (//,10X,'OUTPUT DATA FROM FOUT2',/,
     110X,'MEAN ENERGY VERSUS POSITION')
  503 FORMAT (//,10X,'MEAN ENERGY RATIO')
  617 FORMAT (//,10X,'MEAN ENERGY VS POSITION',/,
     120X,'PLOT TIME =',I5,3X,'NT = ',I5,10X,'TIME = ',2X,1PE10.3,
     22X,'SECONDS')
  717 FORMAT (//,10X,'MEAN ENERGY RATIO',/,20X,'PLOT NUMBER',
     12X,I5,3X,'NT = ',I5,10X,'TIME = ',2X,1PE10.3,2X,'SECONDS')
 1001 FORMAT (30X,'FOUT2 --- CHECK POINT 1 ')
 1002 FORMAT (30X,'FOUT2 --- CHECK POINT 2 ')
 1003 FORMAT (30X,'FOUT2 --- CHECK POINT 3 ')
 1004 FORMAT (30X,'FOUT2 --- CHECK POINT 4 ')
 1005 FORMAT (30X,'FOUT2 --- CHECK POINT 5 ')
 1006 FORMAT (30X,'FOUT2 --- CHECK POINT 6 ')
 6001 FORMAT (/,10X,'TIME STEP VALUE',2X,I3,5X,'MEAN ENERGY = ',
     15X,1PE10.3,2X,'ELECTRON VOLTS')
      RETURN
      END


      SUBROUTINE FOUT3
C        REVISED  2  JUNE  1971
C
C     5 MAY 1971   FIRST DRAFT OF DETECTOR  RESPONSE FUNCTION
C     THIS ROUTINE WILL REQUIRE THE MODIFICATION FOR THE NAMELIST
C     DATA ENTERED IN SUBROUTINE INCON1 TO REFLECT THE ADDITION
C     OF TWO ITEMS OF INFORMATION --- THE MAXIMUM AND MINIMUM ENERGY
C     THAT THE DETECTOR WILL RECOGNOZE
C     WE NOW MUST CONSIDER THE THE MINIMUM INTENSITY THE
C     DETECTOR CAN RESPOND TO-- AS WITH SEVERAL OTHER OF THE
C     ROUTINES , WE WILL SELECT AN ARBITRARY VALUE OF  0.001 OF THE
C     NORMALIZED VALUE
C     THIS NORMALIZED LIMIT IS SELECTED AS A RESULT OF THE AVAILABLE
C     ACCURACY AND PRECESION OF THE CAL COMP PLOTTER WHICH CAN PLOT
C     TO 0.001 OF AN INCH OR SOME VALUE WITHIN THAT RANGE
C     COMMON BLOCKS 3 AND 6  MUST BE REVISED TO HANDLE MORE THAN
C     SIX HARMONIC MODES OF THE FUNCTIONAL DATA INPUT
      REAL ESIG(27)/1.0E07,6.5E06,4.0E06,2.5E06,1.4E06,8.0E05,4.0E05,
     12.0E05,1.0E05,4.65E04,2.15E04,1.0E04,4.65E03,2.15E03,1.0E03,
     24.65E02,2.15E02,1.0E02,4.65E01,2.15E01,10.0,4.65,2.15,1.0,0.465,
     30.215,2.5E-02/
      REAL SIGMAS(26)/1.2,1.65,2.20,3.00,4.10,5.70,8.10,11.0,14.0,
     116.6,18.5,19.3,19.7,20.0,20.1,20.2,20.2,9*20.3/
      REAL LABEL
      REAL LABL(2)/'F(X)','FXT'/
```

```
      REAL*8 TITLE(12)/'MILLS   ','  ','DETECTOR','RESPONSE',' ',' ',
     1'BOX-M',' ','F(X) VS ','T   &  F','(X,T) VS','T      '/
      DIMENSION ATIMX(20),PDUM1(20),PDUM2(20)
      DIMENSION VSIG(150)
      EQUIVALENCE (DUMMY(1),VSIG(1))
      EQUIVALENCE (TXBX(70),ATIMX(1)),(DUMMY(70),PDUM1(1))
      EQUIVALENCE (DUMMY(40),PDUM2(1))
      COMMON/CCM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
      COMMON/CCM7/ S01,S02,S03,S04,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
      COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
      COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
     1NSTAT,MODE
      COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
      COMMON/COM6/ NTOP(6,20),NM(6,20),IKNT(6)
      CALL CANCEL(2)
C     THIS SERIES DEFINES THE PRODUCT-VELOCITY*CROSS SECTION FOR
C     THE CALCULATION OF THE TOTAL FLUX PROFILE:
C     FLUX ( U,X,T) = V(U)*MACRO SIGMA(U)*N ( U,X,T )
C     V( U )  =     VELOCITY OF NEUTRON IN STATE-U
C     SIGMAS(U)=    MACROSCOPIC SROSS SECTION IN BARNS*NUMBER DENSITY
C                   OF HYDROGEN GAS AT STP
C     N(U,X,T)=     NUMBER DENSITY(FRACTIONAL) ON TOTAL NEUTRONS IN
C                   STATE-U, AT TIME-T, AT POSITION-X
      ANZERO= 5.38E19
      DO 228 IP=1,NVIR
      DO 228 IPE=1,26
      IPEP=IPE+1
      IF (E(IP).GT.ESIG(IPE)) GO TO 228
      IF (E(IP).LT.ESIG(IPEP)) GO TO 228
      VSIG(IP)=SIGMAS(IPE)*(1.0E-24)*ANZERO*SQRT((1.918E12)*E(IP))
  228 CONTINUE
C     USE THE LOGICAL CONTROL VARIABLE L(12) TO DEFINE THE TYPE
C     DETECTOR SPECTRUM TO CONSIDER
C     IF L(12) = 1  PLOT F(X,T) VS T
C     IF L(12)=0  PLOT F(X) VS T
C     DETERMINE THE POSITION FUNCTIONS PHIX(NPOINT,NMODE)
C     SET INITIAL VERSION TO CONSIDER 5 POSITION POINTS IN THE
C     DETECTOR SPACE
C     DETECTOR LOCATION IS DEFINED AS A MIDPOINT OF X= S3
C     WITH A HALF WIDTH OF S4
C     SET THE POSITION VARIABLES TXBX(IA) FOR 5 DATA POSITIONS
C     FOR X = S3, S3(+/-).5*S4, S3(+/-)S4
C     USE A FIVE POINT WEIGHTED AVERAGE VALUE FOR THE FINAL RESULT
      DELX=S4/2.0
      DO 4 IXP=1,5
      TXBX(IXP)=(S3-S4)+IXP*DELX
```

```
    4 CONTINUE
      TXBX(3)=S3
C     DETECTOR RESPONSE
C     FOLLOWING DATA MUST BE STATED IN THE BASIC PROBLEM DEFINITION
C     FOR THIS ROUTINE TO RUN SUCCESSFULLY
C        S3         LOCATION OF THE MID POINT OF THE DETECTOR
C        S4         THE HALF WIDTH OF THE DETECTOR
C     SET LOGICAL VARIABLE L( 16 )=1 AND CALL FCUT1 TO RETURN THE
C     VALUES  F(U,T,X) TO THIS ROUTINE FOR EACH POSITION POINT
      L(16) = 1
      NCALL3 = IKNT(1)
      DO 41 IXPNT=1,5
      CALL FOUT1(TXBX(IXPNT))
      DO 42 NTVAL=1,NCALL3
      BX3=0.C
      NUMX= NVIR - 1
      DO 43 NUVAL=1,NUMX
      BX3 = BX3 + FUTX(NUVAL,NTVAL)*VSIG(NUVAL)
   43 CONTINUE
      FXT(IXPNT,NTVAL)=BX3
   42 CONTINUE
   41 CONTINUE
C     RESET L(16) = 0 AFTER CALL TO FOUT1
      L(16)=C
C
C     TYPE OF RESPONSE THAT WE WOULD WISH TO PLOT OR PRESENT
C        F(X,U,T) ------SUM OVER ALL MODES OF DATA
C        F(X,T} -- SUM OVER ALL LETHARGY STATES THAT THE DETECTOR CAN
C                   OBSERVE BETWEEN EMAX AND EMIN
C        F(X,T)     FUNCTIONAL RESPONSE AT A GIVEN TIME STEP
C        F(X)       SUMMED RESPONSE OF THE INTERGRAL/SUM OVER ALL
C                   TIME STEPS AT THE DETECTOR POSITION
C     FUNCTION  F(X,T) IS STORED IN THE ARRAY--FXT(NX,NT)
C     USE ARRAY DUMMY TO STORE THE SUM OVER ALL T FOR F(X, )
C     USE ARRAY TXBX TO STORE DATA FOR INDIVIDUAL TIME STEP DATA F(X,T)
C     VALUE OF NCALL3 IS THE MAX TIME STEP POINT VALUE FROM MOD-5
      CX=0.0
      DO 47 NTVAL=1,NCALL3
      CX1=FXT(1,NTVAL)+FXT(5,NTVAL)
      CX2=FXT(2,NTVAL)+FXT(4,NTVAL)
      CX3=(CX1+3.0*CX2+5.0*FXT(3,NTVAL))/9.0
      DUMMY(NTVAL)=CX3
   47 CONTINUE
C
C     REVISED  18 MAY  1971
C        TIME INTERGRATION SEQUENCE FOR THE DETECTOR RESPONSE FUNCTION
      DX=0.0
      DO 497 NTVAL=2,NCALL3
```

```
      NTMIN=NTVAL-1
      DX1=(DUMMY(NTVAL)+DUMMY(NTMIN))*0.50
      DX2=TIM(NTVAL)-TIM(NTMIN)
      DX3=DX1*DX2
      DX=DX+DX3
      TXBX(NTVAL)=DX
  497 CONTINUE
      TXBX(1)=0.0
C     NOW START THE DATA PLOT TO SHOW THE DETECTOR RESPONSE
C     AS COMPUTED ABOUT THE MIDPOINT OF THE DETECTOR
C     CAL COMP PLOTTER ROUTINE
C     FOUT3 -- DETECTOR RESPONSE FUNCTIONS  -- OUTPUT ROUTINES
C     ALL OUTPUT WILL BE GIVEN BY THE THREE FORMS
C     PRINT ALL VALUES : PLOT ON PRINTER : PLOT USING CAL COMP
C     STANDARD NPS OUTPUT METHODS
      NTCALL = IKNT(1)
C     OUTPUT ON PRINTER
C     PRINT OUT ALL VALUES OF THE TWO FUNCTIONS
C     F (X,T) AND F (X ) VERSUS TIME PRIOR TO DOING THE
C     FINAL PLOT USING THE NPS PLOTTING ROUTINE // DRAW  //
C     USE THE PRINTER AND THE ROUTINE -- PLOT-P TO OBTAIN A
C     PRELIMINARY DATA PLOT
      WRITE (6,154) ((TXBX(IA),TIM(IA)),IA=1,NTCALL)
      MODC = 0
C     13 MAY 1971   ---- SIMPLE FIX-UP  /// REMOVE LATER
C      TEST SERIES TO PLOT DATA DESPITE FUNNY RESULTS ON FIRST RUN
      IKFXT=0
      IKFT=0
      DO 1000 IK6=1,NTCALL
      PDUM1(IK6)=DUMMY(IK6)
      IF (PDUM1(IK6).LT.0.0) IKFXT=IKFXT+1
      PDUM2(IK6)=TXBX(IK6)
      IF (PDUM2(IK6).LT.0.0) IKFT=IKFT+1
 1000 CONTINUE
      IF (IKFXT.LE.2) GO TO 1011
      DO 1010 IK6=1,NTCALL
      PDUM1(IK6)=(-1.0)*PDUM1(IK6)
 1010 CONTINUE
 1011 CONTINUE
      IF (IKFT.LE.2) GO TO 1013
      DO 1012 IK6=1,NTCALL
      PDUM2(IK6)=(-1.0)*PDUM2(IK6)
 1012 CONTINUE
 1013 CONTINUE
C
C     PROVIDE A LOG10 CONVERSION ON THE TIME SCALE TO FIT THE PLOTTER
      SC=2.0
      DO 1015 IK6=2,NTCALL
```

```
      ATIMX(IK6)=SC*(ALOG10(TIM(IK6))+10.0)
 1015 CONTINUE
      ATIMX(1)=0.0
      CALL PLOTP(PDUM2,ATIMX,NTCALL,MODC)
      WRITE (6,155) ((DUMMY(IA),TIM(IA)),IA=1,NTCALL)
      CALL PLOTP(PDUM1,ATIMX,NTCALL,MODC)
C     DEFINE SCALE VARIABLES FOR THE DRAW--CAL COMP PLOTTER
C     OVER RIDE CALL DRAW SEQUENCE UNLESS NOUT = 3
      IF ( NOUT.NE.3) GO TO 1003
      MODX=0
      MODY=0
      IYRT=9
      IXUP=0
      ITYPE=1
      EXSCL=1.0/9.0
      YSCAL=TIM(NTCALL)/14.0
C     PREPARE DATA TO PLOT REFLECTED AND REVERSED AXIS ON THE PLOTTER
      DO 151 KPNT=1,NTCALL
      IF (TXBX(KPNT).LE.0.001) TXBX(KPNT)=0.0
      TXBX(KPNT)=(-1.0)*TXBX(KPNT)
  151 CONTINUE
      IWIDE=9
      IHIGH=15
      IGRID=1
C     PLOT F(X,T) ON THE GRAPH FIRST
      MODC1=1
      CALL DRAW(NTCALL,TXBX,TIM,MODC1,ITYPE1,LABL(1),TITLE,EXSCAL,
     1YSCAL,IXUP,IYRT,MODX,MODY,IWIDE,IWIDE,IHIGH,IGRID,LAST)
C     PLOT F(X) VS TIME
C     PREPARE DATA BY REFLECTING ABOUT Y-AXIS
      DO 152 KPNT2=1,NTCALL
      IF ( DUMMY(KPNT2).LE.0.001) DUMMY(KPNT2)=0.0
      DUMMY(KPNT2)=(-1.0)*DUMMY(KPNT2)
  152 CONTINUE
      ITYPE2=2
      MODC2=3
      CALL DRAW(NTCALL,DUMMY,TIM,MODC2,ITYPE2,LABL(2),TITLE,EXSCL,
     1YSCAL,IXUP,IYRT,MODX,MODY,IWIDE,IHIGH,IGRID,LAST)
 1003 CONTINUE
  154 FORMAT (//,10X,'PRELIMINARY DATA PLOTS',/,10X,'TIME INTERGRATED',
     11X,'FUNCTION--F(X) VS TIME',/,20X,'F ( X )',T35,'TIME-SECONDS',/,
     220(/,18X,1PE10.3,8X,1PE10.3))
  155 FORMAT (//,10X,'PLOT OF DETECTOR RESPONSE FUNCTION',/,
     120X,'F ( X,T ) VS TIME',/,20X,'F(X,T)',T35,'TIME-SECONDS',
     220(/,18X,1PE10.3,8X,1PE10.3))
      RETURN
      END
```

```
      SUBROUTINE FOUT4
C        REVISED  2  JUNE  1971
C
C
C     SUBROUTINE FOUT4 - PREFORMS A DATA CHECK OF EACK INPUT MODE VALUES
C     OF THE NEUTRON DENSITY VECTOR FOR EACH BUCKLING MODE TO TEST
C     ALL VALUES TO DETERMINE THOSE DATA POINTS ABOVE AN ARBITRARY
C     MINIMUM VALUE  FOR THE PARAMETER BX
C     THIS VERSION USES   BX  =  0.001
C     AS A MINIMUM POINT VALUE
C     NEUTRON DENSITY VECTOR IS NORMALIZED TO A VALUE OF 1.00
C     IN PROGRAM   MOD--5
C     FOUT4  INITIALLY SCANS THE INPUT DATA TO OBTAIN A MAXIMUM
C     OF THE POINT VALUES OF THE ARRAY----  FUT(IMODE,NU,NT) FOR ALL
C     POINTS ABOVE THE LIMIT OF THE PARAMETER  BX
C     INITIAL DRAFT OF THIS ROUTINE ( 10 APR 71) WILL PROVIDE A PLOT
C     OF THE INPUT SPECTRUM OF FUT(I,J,K)  VS  NU--LETHERGY STATE FOR EACH
C     MODE OF INPUT DATA
C     INITIAL DRAFT IS BASED ON THE MODEL UTILIZING ONLY A MAXIMUM OF
C     SIX INPUT MODES OF DATA FROM MOD - 5
C     ROUTINE WILL PLOT THE FUNCTION--SC*ALOG10(FUT(IMODE,NU,NT)) VS.
C     LETHARGY VALUE -- U   USING THE CAL COMP PLCTTER
C
      COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX.EMIN,DIFLEN,X
      COMMON/COM7/ SO1,SO2,SO3,SO4,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
      COMMON/CCM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
      COMMON/CCM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
     1NSTAT,MODE
      COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
      COMMON/COM6/ NTOP(6,20),NM(6,20) ,IKNT(6)
C
      REAL*8 RMOD(6)/'MODE-1','MODE-2','MODE-3','MODE-4','MODE-5',
     1'MODE-6'/
      REAL LABEL
      REAL*8 TITLE(12)/'MILLS','BOX--M',' FOUT4','TEST ONE',
     1'  ','  ','  ','FUT(U,T',')  VS  ','U','LETHARGY','  '/
C
C     START OF THE MULTIPLE PROCESSOR MODE TO CLEAN UP INPUT DATA
C     AND PROVIDE GRAPHICAL OUTPUT OF NEUTRON SPECTRUM
C
      SC = 3.0
      BX= 1.0E-03
      NPNTS = NVIR - 3
      ITYPE = 0
      EXSCAL = 1.0
      NMAX = NVIR - 1
```

```
      YSCALE = U(NMAX)/14.0
      IXUP = 0
      IYRT = 0
      MODX = 0
      MODY = 0
      IWIDE = 9
      IHIGH = 15
      IGRID = 1
      DO 300 J=1,MODES
      TITLE(6)= RMOD(J)
      DO 92 NXX = 1,150
   92 DUMMY(NXX) = 0.0
      FX=0.0
      NCALL1 = IKNT(J)
      DO 55 K=2,NCALL1
      NNU=NM(J,K)
C     NNU DEFINES THE MAXIMUM STEP VALUES INPUT FROM MOD 5
      FX = 0.0
      DO 50 NU=1,NNU
   50 IF(FUT(J,NU,K).GT.FX) FX=FUT(J,NU,K)
C     THIS SCAN WILL DETERMINE THE MAX VALUE IN THE INPUT DATA
      CX= BX*FX
C     THIS WILL DETERMINE THE MINIMUM VALUES TO CONSIDER
      WRITE (6,108)  CX,BX,FX,K
  108 FORMAT (//,10X,'CX = ',2X,1PE10.3,2X,'BX = ',2X,1PE10.3,
     12X,'FX = ',2X,1PE10.3,2X,'TIME STEP=',2X,I3)
C
C
      MIL1 = NTOP(J,K)
      MIL2 = NM(J,K)
      MIL3 = MIL1 + MIL2
      DO 51 NU=1,NNU
C
      AX = FUT(J,NU,K)
C     SET MINIMUM VALUE FOR LOG CONVERSION
      IF (AX.LT.BX) AX = BX
      MIL4 = MIL1 + NU
      FUTX(MIL4,K) = SC*ALOG10(AX) + 9.0
C
   51 CONTINUE
C         THIS STEP WILL AUTOMATICALLY ZERO OUT ANY POINT AT THE
C         LOW END OF THE SPECTRUM(HIGH-ENERGY;LOW - LETHARGY) TO MAINTAIN
C         THE CORRECT PROFILE OF THE SPECTRUM
      IF (MIL1.LE.2) GO TO 53
      DO 52 KN = 1,MIL1
   52 FUTX(KN,K) = 0.0
   53 CONTINUE
   55 CONTINUE
```

```
C     START OF THE CALL TO DRAW SERIES
      NCALL1 = IKNT(J) - 1
C      THIS SHOULD REMOVE ERRORS IN DRAW CALL
      LAST = 0
      DO 60 J6 = 2,NCALL1
      IF(J6.EQ.2) MODC = 1
      IF ((J6.GT.2).AND.(LAST.EQ.0)) MODC = 2
      IF (J6.EQ.NCALL1) MODC = 3
      IF (LAST.EQ.2) MODC = 1
      DO 59 K1 = 1,NPNTS
   59 DUMMY(K1) = FUTX(K1,J6)
C
C     DO FINAL DATA CHECK TO INSURE THAT PLOT POINTS ARE WITH
C     IN THE LIMITS OF +0.001  AND  1.0
      DO 73  K2= 1,NPNTS
      IF (DUMMY(K2).GT.9.0) DUMMY(K2) = 9.0
      IF (DUMMY(K2).LT.BX) DUMMY(K2) = 0.0
   73 CONTINUE
C     END OF DATA PLOT CHECK ROUTINE
      LABEL = LTAB(J6)
      LAST = 0
      CALL DRAW(NPNTS,DUMMY,U,MODC,ITYPE,LABEL,TITLE,EXSCAL,
     1YSCALE,IXUP,IYRT,MODX,MODY,IWIDE,IHIGH,IGRID,LAST)
      IF (LAST.EQ.1) GO TO 105
      IF (LAST.EQ.3) GO TO 105
   60 CONTINUE
  105 CONTINUE
      IF (LAST.EQ.1) WRITE(6,106)
      IF (LAST.EQ.3) WRITE (6,107)
  106 FORMAT (//,10X,'LAST = 1',3X,'RECHECK DATA')
  107 FORMAT (//,10X,'LAST=3',3X,'RECHECK DATA')
C
C
C
  300 CONTINUE
      RETURN
      END


      SUBROUTINE FOUT5
C     REVISED  15  JUNE  1971
C
C        SUBROUTINE FOUT5 PROVIDES A FOURIER POSITION FUNCTION
C        WEIGHTED SPECTRAL PLOT OF THE INPUT DATA FROM MOD-5 AND
C        PLOTS F(IMODE,U,T)*PHI(IMODE,X1) VS U(LETHAGY) FOR EACH
C        MODE OF INPUT DATA AND A SUMMED SPECTRUM OF ALL MODES OF INPUT
C     L ( 4 ) = 1 DEFINES THE EXCEPTIONAL CASE WHERE THE USER WOULD
C     DESIRE TO PLOT THE SPECTRAL RESPONSE FUNCTION WEIGHTED BY THE
```

```
C      FOURIER EXPANSION COEFFICIENT FOR ECH MODE AND POSITION OTHER
C      TNAN AT THE DEFAULT CASE OF X = 0.0
C      TOTAL POINTS ( POSITION ) THAT THE PROGRAM WILL CONSIDER IS
C      INPUT USING THE CONTROL VARIABLE NANS ( 6 ) FOR THE NUMBER OF
C      POINTS AND THE INPUT ARRAY XBX( I ) FOR THE ACTUAL POINTS
C      PRE CALCULATE THE  FOURIER SPACE DEPENDENT VALUES FOR EACH MODE
C      FOR EACH POINT X1
       COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
       COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
       COMMON/COM7/ SO1,SO2,SO3,SO4,AK11,AK12,AK13,AK14,AN3A,AN3B,
      1AN3C,ANC4,ANA4,ANB4,AN2X,AK15,AK5X
       COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
       COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
      1NSTAT,MODE
       COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
       COMMON/CCM6/ NTOP(6,20),NM(6,20),IKNT(6)
       REAL*8 RTITLE(12)/'MILLS','BOX  M','  ','FOUT5','  ','  ',
      1'TOTAL','SUM OF','ALL','MODES','  ','  '/
       REAL*8 ANAME(12)/'MILLS','BOX M','FOUT5','  ','  ','  ','  ','  ',
      1'  ','  ','  ','  '/
       REAL*8 BMODE(5)/' 1 - 2','1-2-3','1-2-3-4','12345','123456'/
       REAL*8 APNT(9)/'POINT- 1','POINT- 2','POINT- 3','POINT- 4',
      1'POINT- 5','POINT- 6','POINT- 7','POINT-8','POINT- 9'/
       REAL*8 AMODE(6)/'MODE-1','MODE-2','MODE-3','MODE-4','MODE-5',
      1'MODE-6'/
       REAL LABEL
       DIMENSION PHIMX(6)
C      START EXECUTION  DEFINE PARAMETERS--SCALE FACTORS,ETC FOR EXEC
C
       IPNTR=NANS(2)
       IF ((IPNTR.LE.0).OR.(IPNTR.GE.9))  IPNTR=9
       RTITLE(12)=APNT(IPNTR)
       ANAME(12)=APNT(IPNTR)
       NRUNS  = NANS(6)
C      THIS DEFINES THE TOTAL NUMBER OF DATA POINTS THAT WILL
C      HAVE STECTRAL RESPONSE DATA POINTS PLOTTED
       IF (( NRUNS.LE.0).OR.(NRUNS.GE.3)) GO TO 5500
       IF (L(4).EQ.0) NRUNS = 1
       DO 5001 NTIMX = 1,NRUNS
       X1= XBX(NTIMX)
       IF ( L(4).EQ.0) X1=0.0
       WRITE (6,1013) X1
  1013 FORMAT (//,10X,'POSITION POINT FOR SPECTRAL PLOT =',5X,F8.2)
       DO 1011 JXX= 1,MODES
       PHIMX(JXX)=PHI1(JXX,IS,X1)
       WRITE(6,1012) JXX,PHIMX(JXX)
  1012 FORMAT (//,10X,'MODE =',2X,I5,'PHIX = ',E15.8)
  1011 CONTINUE
```

```
C
      SC = 3.0
      ITYPE = 0
      EXSCAL = 1.0
      NPNTS = NVIR-1
C     TEST SERIES CORRECTION FOR MIN SCALING VALUE TO PLOT
C     THIS SHOULD BE ADJUSTED TO BECOME PART OF THE INPUT
C     NAMELIST DATA-- DATAIN  -- TO ALLOW THE USER TO SELECT
C     AN ESTIMATED BEST VALUE ----  THIS WILL REQUIRE ADDITION
C     TO ONE OF THE PRINCIPAL COMMON BLOCKS TO TRANSFER
C     PLUS THE ESTABLISHMENT OF A DEFAULT OPTION IN INCON1
C     REVISED  15  JUNE  1971
      BX=1.0E-05
      YSCALE = U(NPNTS)/14.0
      IXUP = 0
      IYRT = 0
      MODX = 0
      MODY = 0
      IWIDE = 9
      IHIGH = 15
      IGRID = 1
      DO 92 NXX=1,150
      DO 92 NTT = 1,20
      FUTX(NXX,NTT) = 0.0
   92 CONTINUE
      DO 300  IX=1,MODES
      ANAME(6) = AMODE(IX)
      NCALL1 = IKNT(IX)
      DO 55 K=2,NCALL1
      DO 49 IXX = 1,150
   49 DUMMY(IXX) = 0.0
      FX= 0.0
      MIL1 = NTOP(IX,K)
      MIL2 = NM(IX,K)
      MIL3 = MIL1 + MIL3
      DO 51 NU = 1,MIL2
      MIL4 = MIL1 + NU
      AX = PHIMX(IX)*FUT(IX,NU,K)
      FUTX(MIL4,K) = AX + FUTX(MIL4,K)
      IF ( AX.LE.BX)  AX = BX
C        PLOTTING FACTOR ADJUSTMENT
      CX=SC*ALOG10(AX) + 15.0
      IF ( CX.GT.9.0) CX = 9.0
C
      IF ( CX.LT.BX)  CX=0.0
      DUMMY(MIL4) = CX
   51 CONTINUE
      LABEL = LTAB(K)
```

```
      IF ( K.EQ.2) MODC = 1
      IF (( K.GT.2).AND.(LAST.EQ.0)) MODC = 2
      IF ( K.EQ.NCALL1) MODC = 3
      LAST = 0
      CALL DRAW(NPNTS,DUMMY,U,MODC,ITYPE,LABEL,ANAME,EXSCAL,YSCALE,
     1IXUP,IYRT,MODX,MODY,IWIDE,IHIGH,IGRID,LAST)
      IF(LAST.EQ.1) GO TO 105
      IF (LAST.EQ.3) GO TO 105
   55 CONTINUE
  105 IF(LAST.EQ.1) WRITE (6,106)
      IF (LAST.EQ.3) WRITE (6,107)
  300 CONTINUE
      LAST = 0
C     SECOND PART OF THE DRAW PLOT ROUTINE TO PLOT THE SUM OF THE TOTAL
C     HARMONIC EXPANSION OF THE NEUTRON DENSITY FUNCTIONS
      NKNT = IKNT(1)
      MX=MODES-1
      IF(MX.EQ.0) MX=1
      RTITLE(6) = BMODE(MX)
      DO 310 NXT = 2,NKNT
      DO 309 NTNX =1,NPNTS
      AX = FUTX(NTNX,NXT)
      IF(AX.LT.BX) AX = BX
      DUMMY(NTNX) = SC*ALOG10(AX) + 9.0
  309 CONTINUE
      LABEL = LTAB(NXT)
      IF(NXT.EQ.2) MODC = 1
      IF ((NXT.GT.2).AND.(LAST.EQ.0)) MODC = 2
      IF (NXT.EQ.NKNT) MODC = 3
      LAST = 0
C     PROTECTION STEPS TO PREVENT  EXCESSIVE PLOTTING ON THE CAL COMP
C     TO OVER RIDE THIS PROTECTION,THE ROUTINE MUST BE REWRITTEN
      NFLAG=NFLAG+1
      IF (NFLAG.GE.4) GO TO 5500
      CALL DRAW(NPNTS,DUMMY,U,MODC,ITYPE,LABEL,RTITLE,EXSCAL,YSCALE,
     1IXUP,IYRT,MODX,MODY,IWIDE,IHIGH,IGRID,LAST)
      IF(LAST.EQ.1) GO TO 320
      IF(LAST.EQ.3) GO TO 320
  310 CONTINUE
  320 IF(LAST.EQ.1) WRITE(6,106)
      IF(LAST.EQ.3) WRITE (6,107)
  106 FORMAT (//,10X,'LAST = 1',3X,'RECHECK DATA')
  107 FORMAT (//,10X,'LAST=3',3X,'RECHECK DATA')
 5001 CONTINUE
 5500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE READ11(LASTN)
C     REVISED  5  MAY  1971
C        THIS SUBROUTINE HAS NOT BEEN COMPLETELY TESTED AND DEBUGGED
C
C     READ11 INPUTS THE SAME TEST DATA AS ROUTINE READ10
C     BY PROCESSING NINE(9) TRACK MAGNETIC TAPE
C     INITIALLY AT A DENSITY OF 800 BPI
C     IDENTICAL INPUT/OUTPUT MESSAGES AND DATA ARE FURNISHED
C     AS BY READ10
C
      REAL MTITLE (54)
      DIMENSION NNRUN(20),NNT(20)
C
      COMMON/COM1/ XBX(100),U(150),E(150),TIM(20),DUMMY(150),TXBX(100)
      COMMON/COM4/ SOH,SA,S1,S2,S3,S4,SZERO,BUCKLE,EMAX,EMIN,DIFLEN,X
      COMMOM/COM7/ SO1,SO2,SO3,SO4,AK11,AK12,AK13,AK14,AN3A,AN3B,
     1AN3C,ANC4,ANA4,ANB4,AN2X
      COMMON/COM2/ NT(20),NPROB(54),NANS(6),L(20),LTAB(20)
      COMMON/COM5/ MODES,IS,IDATA,N,NVIR,NI,NF,NCALL,NOUT,IBX,
     1NSTAT,MODE
      COMMON/COM3/ FXT(100,20),FUTX(150,20),BUCK(6),FUT(6,71,20)
      COMMON/COM6/ NTOP(6,20),NM(6,20),IKNT(6)
C
C     SET DEFAULT OPTION FOR LASTN = 0
      LASTN=0
      DO 8 ISET=1,MODES
      READ (4,20,END=8,ERR=9) (MTITLE(IA),IA=1,54)
      READ (4,21,END=8,ERR=9) N,NVIR,NI,NF,BUCK(ISET)
      IF (ISET.GT.1) GO TO 1
      READ (4,22) (U(IA),IA=1,NVIR)
      READ (4,22) (E(IA),IA=1,NVIR)
    1 NCOUNT = 0
      NCALL1 = 0
    2 NCALL1 = NCALL1 + 1
      IF (NCALL1.GE.21) GO TO 18
      READ (4,23,END=8,ERR=9) IRUN,TIM(NCALL1),NNT(NCALL1),
     1NTOP(ISET,NCALL1),NM(ISET,NCALL1)
      NNRUN(NCALL1) = IRUN
      MIL1 = NM(ISET,NCALL1) + 1
      READ (4,24,END=8,ERR=9) (FUT(ISET,IB,NCALL1),IB=1,MIL1)
      IF (IRUN.EQ.1) GO TO 2
C     END OF TAPE READ IN LOOP FOR NORMAL TIME STEP INPUT DATA
C
C
C     DEFAULT OPTION IF MORE THAN 20 DATA SETS IS PRODUCED PER
C     TINE STEP IN MOD 5
   18 WRITE (6,27)
      NKNT = 0
```

```
   19 READ (4,23,ERR=9) IXRUN,XTIM,NXNT,NXNM
      NKNT = NKNT + 1
      MIL2 = NXNM + 1
      READ (4,24,END=8,ERR=9) (DUMMY(IA),IA=1,MIL2)
      WRITE (6,28) IXRUN,XTIM,NXNT,NXTOP,NXNM
      IF(NKNT.GT.20) GO TO 9
C     PROTECTION STEP TO GET OUT OF INFINITE TAPE READ LOOP
      IF (IXRUN.EQ.1) GO TO 19
      WRITE(6,29) NKNT
C     END OF TAPE READ DEFAULT OPTIONS
C     DATA PRINT OUT ROUTINE TO INFORM USER OF DATA SETS INPUT
      IF (ISET.GT.1) GO TO 4
      DO 3 IX=1,NCALL1
    3 NT(IX) = NNT(IX)
    4 WRITE (6,25) (MTITLE(IA),IA=1,54)
      WRITE (6,26) N,NVIR,NI,NF,BUCK(ISET)
      IF (ISET.GT.1) GO TO 6
      NPX= (NVIR/2) + 1
      DO 5 IK1 = 1,NPX
      IK1P = NPX + IK1
    5 WRITE (6,30) IK1,U(IK1),E(IK1),IK1P,U(IK1P),E(IK1P)
    6 WRITE (6,31)
      DO 7 IX = 1,NCALL1
    7 WRITE (6,29) IX,TIM(IX),NNT(IX),NTOP(ISET,IX),NM(ISET,IX)
      WRITE (6,32)
    8 CONTINUE
      WRITE (6,34)
      GO TO 50
    9 LASTN=1
   50 CONTINUE
   20 FORMAT (18A4)
   21 FORMAT (4I5,E15.8)
   22 FORMAT (8E10.3)
   23 FORMAT (I5,E15.8,3I5)
   24 FORMAT (8E10.3)
   25 FORMAT (3(10X,18A4,/))
   26 FORMAT (/,10X,'N = ',9X,I5,/,10X,'NVIR =',6X,I5,/,10X,'NI =',
     18X,I5,/,10X,'NF = ',8X,I5,/,10X,'BUCKLE = ',3X,F10.5)
   27 FORMAT (///,10X,'TOTAL INPUT TIME DATA FROM MOD-5',//,
     110X,'EXCEEDS 20 DATA SETS')
   28 FORMAT (/,10X,I5,E10.3,3I5)
   29 FORMAT (///,10X,'TOTAL EXCESS DATA SETS =',2X,I5)
   30 FORMAT (10X,2(I3,5X,2(1PE10.3,5X)))
   31 FORMAT (10X,'INPUT DATA FROM MOD 5',//,
     13X,'SET',3X,'TIME',3X,'NT',3X,'NTOP',2X,'NM')
   32 FORMAT (//,10X,3(3X,'****',3X),/,10X,'END OF INPUT DATA SET',/,
     110X,3(8X,'****',3X))
   34 FORMAT(/,10X,'END OF TAPE READ CYCLE')
```

```
RETURN
END
```

BIBLIOGRAPHY


1. Tait, J. H., An Introduction to Neutron Transport Theory, p. 16-23, American Elesevier, 1964.

2. Davison, B., Neutron Transport Theory, p. 25-52, Oxford, 1958.

3. Galanin, A. D., Thermal Reactor Theory, p. 25-54, Pergamon, 1960.

4. Ibsen, H. S., Introductory Nuclear Reactor Theory, p. 389-422, Reinhold, 1963.

5. Gayther, D. B., and Goode, P. D., "Measurement of Fast Neutron Spectra in Reactor Materials," Pulsed Neutron Research, v. 2, p. 435-442, International Atomic Energy Agency, 1965.

6. Lamarsh, J. R., Introduction to Nuclear Reactor Theory, p. 152-185, Addison-Wesley, 1966.

7. Ash, M., Nuclear Reactor Kinetics, p. 152-185, McGraw Hill, 1965.

8. Perkel, D. H., Transactions-American Nuclear Society, 3, 2, 342, 1960.

9. Ohanian, M. J., and Daitch, P. B., "Eigenfunction Analysis of Thermal Neutron Spectra," Nuclear Science and Engineering, 21, p. 227-239, 1964.

10. Jenkins, J. D., and Daitch, P. B., "Analysis of Pulsed Fast Neutron Spectra in Multiplying Assemblies," Nuclear Science and Engineering, 37, p. 41-58, 1969.

11. Williamson, T. J., and Albrecht, R. W., "A Discrete Stochastic Model for Neutron Moderation," Nuclear Science and Engineering, 37, p. 222-233, 1968.

12. Menzel, A. D., and others, "Time-Dependent Neutron Spectra Measurements in Water," Nuclear Science and Engineering, 42, p. 119-136, 1970.

13. Williamson, T. J., and Albrecht, R. W., "A Stochastic Calculation of Fast Reactor Generation Times," Fast Reactor Physics, v. 1, p. 513-526, International Atomic Energy Agency, 1968.

14. Kemeny, J. G., and Snell, J. L., Finite Markov Chains, p. 147-169, van Nostrand, 1969.

15. Abawjan, L. P., and others, Fast Neutron Cross Sections, Pergamon, 1963.

INITIAL DISTRIBUTION LIST

| | | No. Copies |
|---|---|---|
| 1. | Defense Documentation Center<br>Cameron Station<br>Alexandria, Virginia 22314 | 2 |
| 2. | Library, Code 0212<br>Naval Postgraduate School<br>Monterey, California 93940 | 2 |
| 3. | Asst Professor T. J. Williamson, Code 61Wn<br>Department of Physics<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 4. | MAJ Thomas M. Mills, USA<br>345 82nd Street<br>Brooklyn, New York 11209 | 1 |
| 5. | Chief of Personnel Operations<br>Attn: OPD-OPEN<br>Department of the Army - STOP 106<br>Washington, D. C. 20315 | 1 |

UNCLASSIFIED
Security Classification

## DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

| | |
|---|---|
| 1. ORIGINATING ACTIVITY (Corporate author)<br>Naval Postgraduate School<br>Monterey, California 93940 | 2a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED<br>2b. GROUP |

3. REPORT TITLE

SPACE DEPENDENT MODEL FOR THE SLOWING DOWN OF FAST NEUTRONS

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; June 1971

5. AUTHOR(S) (First name, middle initial, last name)

Thomas M. Mills

| | | |
|---|---|---|
| 6. REPORT DATE<br>June 1971 | 7a. TOTAL NO. OF PAGES<br>151 | 7b. NO. OF REFS<br>15 |
| 8a. CONTRACT OR GRANT NO.<br>b. PROJECT NO.<br>c.<br>d. | 9a. ORIGINATOR'S REPORT NUMBER(S)<br>9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) | |

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

| | |
|---|---|
| 11. SUPPLEMENTARY NOTES | 12. SPONSORING MILITARY ACTIVITY<br>Naval Postgraduate School<br>Monterey, California 93940 |

13. ABSTRACT

The slowing down of fast neutrons was analyzed by a multi-group method of discrete time and energy states coupled with a spatial harmonic expansion method to determine the neutron density in a homogeneous, isotropically scattering slab. Five neutron source geometries were studied for both a fissioning and a non-fissioning system.

Numerical results were obtained for the neutron flux, mean neutron energy and the neutron spectra for the one dimensional system using a harmonic mode expansion of up to six terms to determine the time-energy-space dependence.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811



UNCLASSIFIED
Security Classification

A-31408

| KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Neutron Diffusion | | | | | | |
| Boltzman Transport | | | | | | |
| Neutron Slowing Down | | | | | | |
| Multi-Group | | | | | | |

128362

Thesis
M597 Mills
c.1 Space dependent
model for the slowing
down of fast neutrons.